SIT772

Database Fundamentals

THI THU SUONG NGO
STUDENT ID: 224757434

Self-Assessment Details
The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment				✔

Self-Assessment Statement

Based on my comprehensive and consistent demonstration of mastery across all Unit Learning Outcomes, I am applying for a High Distinction in SIT772 - Database Fundamentals, Throughout the trimester, I engaged deeply with the course content, extending my learning beyond the core curriculum through critical reflection, independent research, and application of best practices in database design and management.

This portfolio reflects not only successful completion of the required tasks but also evidence of higher-order thinking, integration of contemporary technologies, and strong ethical awareness in handling data. My ability to link theory to practice, critically evaluate my work, and communicate insights effectively aligns with both the Unit Learning Outcomes and Graduate Learning Outcomes, evidencing a level of performance that meets the expectations of a High Distinction.

Checklist	Included
Learning Summary Report	✔
All tasks required for the target grade completed	✔
Evidence of any additional task(s) or activities completed	✔

Declaration

Learning Summary Report

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: 

Portfolio Overview

This portfolio demonstrates comprehensive achievement of all Unit Learning Outcomes (ULOs) for SIT772- Database Fundamentals to a High Distinction level. It showcases a balanced progression across theoretical understanding, hands-on database work, and research-based critical thinking. A key highlight of my portfolio is the Task 3.2HD research report, where I investigated the integration of machine learning (ML) and data systems, offering insights that significantly deepened my understanding of modern database practices and their intersection with intelligent systems.

Over the course of SIT772- Database Fundamentals, I have progressed from foundational knowledge of databases to mastering advanced skills in relational design, SQL implementation, procedural programming, and research-informed data architecture. My portfolio demonstrates not only the successful application of all five Unit Learning Outcomes (ULOs) but also clear evidence that I have extended well beyond the unit material, reflecting the standard of a High Distinction.

I began the trimester with basic familiarity with databases- comfortable with terms like "tables" and "queries" but lacking practical experience in design, normalization, or system integration. Through the early tasks, I built a strong theoretical foundation by engaging critically with the concepts of data, information, and knowledge, and understanding the limitations of traditional file systems in contrast to relational databases. This understanding shaped the way I approached every subsequent task: not as isolated exercises, but as components of a larger data management ecosystem.

A key turning point was learning to analyse business requirements and model them effectively using Entity Relationship Diagrams. I applied normalization techniques confidently- up to Third Normal Form- to design efficient, scalable schemas that reduced redundancy and preserved data integrity. The real-world complexity of these tasks challenged me, especially in identifying dependencies and structuring many-to-many relationships, but I overcame them through iterative refinement and targeted self-learning.

I advanced rapidly in SQL, moving from basic SELECT queries to complex multi-table JOINs, nested subqueries, data transformation, and constraint enforcement. I implemented full relational databases using MySQL and Oracle, demonstrating fluency in both environments. My work with PL/SQL- including stored procedures, triggers, and functions- went beyond the expected learning outcomes and allowed me to simulate realistic business logic within the database environment.

Where I truly believe I exceeded expectations was in my research report on the intersection of machine learning and modern database systems. This project was self-driven, involving academic literature, industry white papers, and practical case studies. I investigated cutting-edge topics such as in-database model serving, scalable data pipelines (ETL vs ELT), federated learning, and ML-aware data governance. By critically evaluating the infrastructural needs of intelligent systems, I gained a perspective that allowed me to reframe my database work- not as standalone design- but as the foundation of future-ready AI ecosystems.

In addition, I extended my learning through data visualization and BI tools by building PivotCharts in Excel and designing a data-driven form interface using web technologies and embedded SQL. These tasks, though not central to the core unit, demonstrate my initiative to explore integration points between databases and user interaction, and to think holistically about the end-to-end data lifecycle.

Throughout the trimester, I approached each task with the mindset of continuous improvement. I actively engaged with feedback, revised submissions, and sought out advanced resources to refine my understanding. I overcame challenges in form validation, relational modelling, and PL/SQL debugging not through trial and error alone, but by drawing on documentation, peer collaboration, and external tutorials- demonstrating both resilience and self-direction.

Each task in this portfolio aligns strongly with the Unit Learning Outcomes:

- ULO1: Demonstrated through foundational knowledge and critical comparison of traditional and modern data systems.
- ULO2: Applied through professional ERD design and normalization grounded in real-world scenarios.
- ULO3: Evidenced by robust, multi-table database implementation using SQL and procedural logic in PL/SQL.
- ULO4: Exceeded through original research on machine learning infrastructure and advanced data architecture.
- ULO5: Reflected in self-assessment, integration of feedback, and thoughtful synthesis across tasks.

I believe this portfolio reflects not only technical excellence and comprehensive coverage of the unit learning outcomes, but also the curiosity, discipline, and critical engagement expected at the High Distinction level. I have extended beyond the material, demonstrated initiative in bridging database practice with emerging technologies, and maintained a high standard of academic and professional integrity throughout.

I respectfully submit this portfolio as evidence of my capability, commitment, and readiness to contribute meaningfully to the data and technology landscape- and ask that it be considered for a High Distinction.

Reflections

The most important things I learnt:

This unit taught me that designing, implementing, and managing a database is not just a technical process but a structured and strategic discipline. I learnt how to approach real-world problems methodically- starting with identifying data requirements, developing normalized data models, implementing relational databases, and finally applying SQL for data manipulation and control. Importantly, I learnt to view databases not just as static systems but as dynamic components of larger architectures, particularly when integrated with business intelligence or machine learning systems. This perspective shift was a key takeaway that exceeded my original expectations.

I feel I learnt these topics, concepts, and/or tools really well:

I now feel confident in data normalization (1NF, 2NF, 3NF), writing complex SQL queries involving multiple joins and subqueries, designing ERDs, and implementing constraints and relationships in a relational schema. I also became proficient with tools like MySQL Workbench, Oracle PL/SQL, and Excel PivotTables. Moreover, I feel comfortable explaining the logic and trade-offs behind data modelling decisions, which has improved both my technical skills and communication confidence.

I found the following topics particularly challenging:

The most challenging part of the unit was working with procedural SQL in Oracle during Task 8.2D. Setting up the Oracle environment, understanding the documentation, and implementing stored procedures, triggers, and functions all presented a steep learning curve. Initially, I found the syntax complex, the error messages unclear, and the logic difficult to debug.

However, through perseverance, repeated practice, and actively seeking external resources, I gradually developed confidence and clarity. I now have a solid understanding of how PL/SQL operates and how procedural logic can be embedded within database systems. More importantly, this challenge taught me a lot about my learning process. I discovered that I learn best through hands-on experimentation, breaking down complex tasks into smaller parts, and staying resilient even when progress feels slow. Overcoming this hurdle gave me a renewed sense of confidence in my ability to tackle unfamiliar technologies independently.

I found the following topics particularly interesting

One of the most interesting and valuable parts of this unit was learning how to embed SQL logic into an interactive user interface, as I did in Task 9.2D. Designing a Windows Forms application that connects to a MySQL database helped me understand how database systems operate in real-world software environments- beyond writing queries in a console or editor.

It was fascinating to see how input validation, constraint handling, and SQL insertion commands come together in a user-facing application. Building the form from scratch, handling events, and using parameterized SQL to prevent injection gave me a deeper appreciation of the security and usability considerations involved in data systems. More importantly, this task bridged my database skills with my interest in software development, reinforcing my understanding of how front-end and back-end systems integrate.

This task helped me realize how much I enjoy building practical tools that interact with databases- something I hadn't fully appreciated before. It sparked my interest in pursuing more work in full-stack development or data-driven applications, where I can design solutions that are both technically sound and user-focused.

I still need to work on the following areas:

While I have gained a strong foundation in relational database systems, I recognize there are still areas where I can improve and grow further. In particular, I want to deepen my understanding of performance optimization techniques- such as indexing strategies, query execution plans, and load balancing in multi-user environments. These are advanced topics that I encountered conceptually but have not yet fully explored in practice.

I also identified that I could improve my efficiency in handling unfamiliar environments. Tasks involving Oracle PL/SQL and system setup required considerable time navigating documentation and troubleshooting configuration issues. In future units, I aim to become more confident and systematic in approaching new technologies by building a stronger foundation in reading technical documentation and practicing with more diverse platforms.

Finally, from a personal development perspective, I intend to continue refining my time management and task estimation skills. While I maintained consistent progress overall, I found that allocating more time for extended tasks- such as research and implementation projects- would allow for deeper reflection and iteration.

As a next step, I plan to build on the knowledge gained in this unit by exploring cloud-based data systems, big data platforms, and data-driven application development, aligning with my long-term interest in intelligent systems and data engineering. This unit has provided the confidence and motivation to continue growing as a learner and aspiring IT professional.

The things that helped me most were:

The most valuable resources throughout this unit were a combination of structured learning materials, academic support, and self-directed exploration. I regularly reviewed the lecture slides on Deakin Cloud, which provided concise and well-organized summaries of core concepts. These slides were especially helpful when revisiting complex topics like normalization, transaction control, and data security, and gave me a reliable reference throughout the trimester. The workshop walkthroughs and worked examples made abstract ideas- such as ERD design, relational constraints, and advanced SQL queries- more accessible and easier to apply.

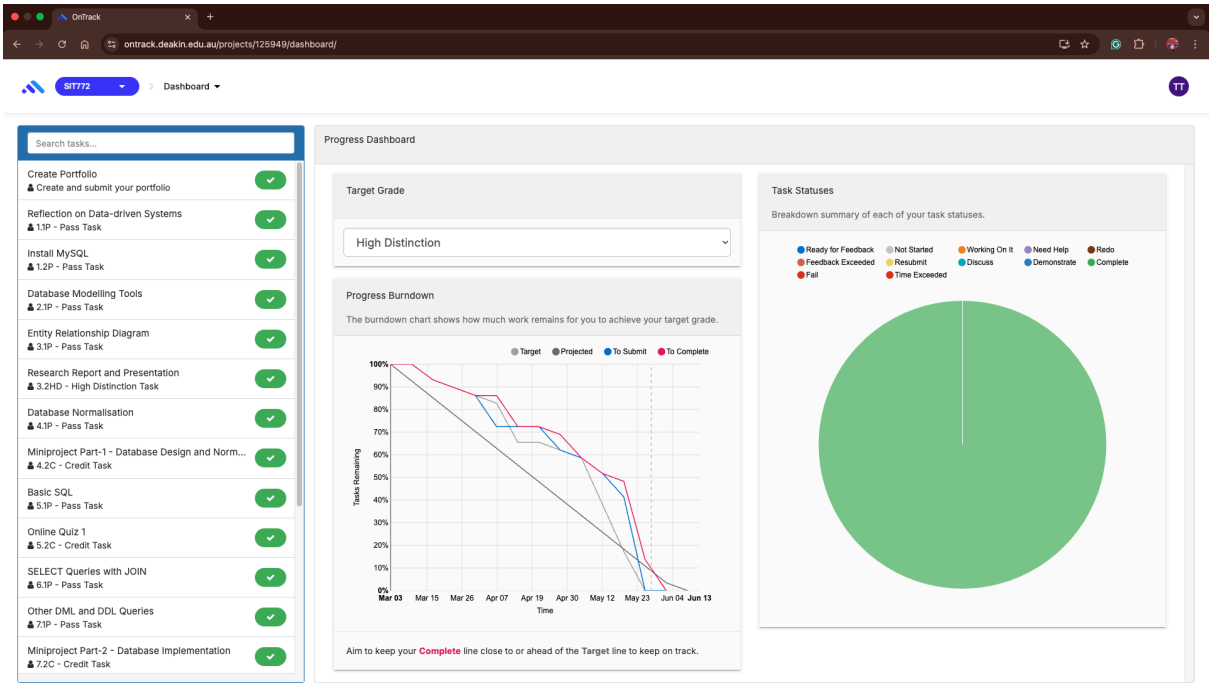
The OnTrack platform further supported my progress with clear weekly task details and supported me to improve my technical skills and improve deadline management.

Beyond this, the lecturer's explanations delivered through classes played a crucial role in helping me grasp concepts like functional dependencies and referential integrity. Concepts were often contextualized using relatable, real-world analogies, which helped bridge theory with practical application.

I also benefited from the tutoring sessions, where I had opportunities to ask questions, resolve issues, and gain timely feedback. These sessions boosted my confidence, especially when working through Oracle PL/SQL and during the independent research component of my HD task.

Finally, learning to navigate official documentation (such as MySQL and Oracle references) and engaging with academic and industry sources during my research project helped me think more critically and become a more independent learner. It was this combination of flexible cloud-based learning, strong academic guidance, and self-driven inquiry that made the unit both effective and rewarding.

My progress in this unit was …:



My progress in this unit was steady, intentional, and goal-focused. The OnTrack burndown chart reflects a consistent commitment to completing tasks over time, with all tasks completed well before the deadline. I maintained a trajectory that closely followed the projected target line, which shows that I took a proactive approach to managing my workload rather than rushing at the last minute.

By consistently aiming to stay ahead of schedule, I gave myself the space to revise, reflect, and seek feedback- especially on complex or distinction-level tasks such as the PL/SQL implementation (8.2D), embedded user interface (9.2D), and the research report and presentation (3.2HD). This approach allowed me to explore the material more deeply and extend my learning beyond the unit expectations.

The progress pie chart further shows that all of my tasks were completed successfully. This result reflects not only effective time management but also a clear strategy in how I tackled progressively more advanced tasks as the unit developed. Overall, the progress data reinforces my ability to self-manage, plan ahead, and engage meaningfully with the learning process- all qualities that support my application for a High Distinction.

If I did this unit again, I would do the following things differently:

Reflecting on this unit, one thing I would do differently is to seek more intentional and meaningful engagement in class and offer support to peers when I could, I found that these efforts sometimes felt surface-level or didn't lead to deeper insight. If I had approached these moments with more focus- such as preparing thoughtful questions, offering peer support that challenged both of us to think critically, or initiating discussions about advanced topics- I believe I could have gained more from the collective learning experience.

I also extended my learning beyond the unit, particularly through my research project, which connected machine learning and database infrastructure. However, looking back, I now realise I could have gone further by applying those insights in more practical ways, such as experimenting with cloud-based ML pipelines or prototyping integration with database systems. The learning was there- but the impact could have been stronger had I pushed myself to apply it more boldly.

This unit helped me see that it's not just about what you do beyond the core requirements, but how deeply and intentionally you do it. In future units, I want to focus less on the quantity of extra effort and more on the quality and depth of engagement- whether that's through stronger contributions in class, more impactful peer collaboration, or more applied experimentation outside of assessments.

Other…:

One of the most valuable outcomes of this unit has been the shift in how I see myself- not just as a student completing tasks, but as a growing IT professional who builds systems with real-world purpose, long-term impact, and ethical responsibility. Through designing normalized data models, writing secure and structured SQL solutions, and engaging in hands-on implementation, I've developed a deeper understanding of how databases operate at the heart of modern information systems.

This growth was especially solidified through my High Distinction research project, where I explored the integration of machine learning and database infrastructure. That task pushed me far beyond the standard expectations, requiring me to critically evaluate tools like MLflow and BigQuery, assess architectural trade-offs, and reflect on emerging concerns like data drift, privacy, and ethical model deployment. It strengthened my ability to work independently, synthesise academic and industry sources, and contribute to conversations about the future of intelligent systems.

This unit also reinforced the importance of aligning technical skills with broader values: data ethics, scalability, usability, and maintainability. I learned that successful systems aren't just functional- they are sustainable, adaptable, and thoughtfully designed to meet evolving needs. Whether I was writing a PL/SQL trigger, building a WinForms-based embedded SQL interface, or supporting peers with explanations, I was constantly encouraged to think critically about both how and why we build systems the way we do.

Perhaps most importantly, SIT772 reminded me that meaningful growth comes not only from meeting requirements, but from the choices we make to explore beyond them. The unit gave me confidence in my ability to independently learn new tools and frameworks, and a mindset focused on continuous learning, real-world relevance, and thoughtful contribution. These lessons have helped define my professional direction, and I carry them forward with clarity and purpose into future roles in data engineering, full-stack development, or intelligent systems architecture.

Deakin University

Database Fundamentals

Thi Thu Suong Ngo

Portfolio Submission

Submitted By:
Thi Thu Suong Ngo
s224757434

Tutor:
Iynkaran Natgunanathan

June 5, 2025

Contents

2 Overall Task Status

Task	Status	Times Assessed
Reflection on Data-driven Systems	Complete	1
Install MySQL	Complete	1
Database Modelling Tools	Complete	1
Entity Relationship Diagram	Complete	1
Research Report and Presentation	Complete	4
Database Normalisation	Complete	2
Miniproject Part-1 - Database Design and Normalisation	Complete	2
Basic SQL	Complete	1
Online Quiz 1	Complete	1
SELECT Queries with JOIN	Complete	1
Other DML and DDL Queries	Complete	1
Miniproject Part-2 - Database Implementation	Complete	1
Online Quiz 2	Complete	2
PL/PSQL - Trigger, Procedure and Function	Complete	3
Data Analysis and Visualization using Excel	Complete	1
Interaction with a database via an User Interface	Complete	2
Database security	Complete	1
Learning Summary Report	Complete	1

3 Learning Outcomes

3.1 Fundamental concepts of database

Interpret and explain fundamental concepts of data, information, and knowledge and demonstrate an understanding of differences between traditional file systems and databases.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦♦♦	Complete	1
Database Modelling Tools	♦♦♦♦♦	Complete	1
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	4
Database Normalisation	♦♦♦♦♦	Complete	2
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	2
Basic SQL	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	2
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	3
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	2
Learning Summary Report	♦♦♦♦♦	Complete	1
Database security	♦♦♦♦♦	Complete	1

3.2 Relational Database Modelling

Analyse real-world problems to identify data requirements and apply data modelling concepts to design and develop Entity Relationship Diagrams for efficient data representation and storage.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦♦♦	Complete	1
Database Modelling Tools	♦♦♦♦♦	Complete	1
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	4
Database Normalisation	♦♦♦♦♦	Complete	2
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	2
Basic SQL	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	2
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	3
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	2
Learning Summary Report	♦♦♦♦♦	Complete	1
Database security	♦♦♦♦♦	Complete	1

3.3 Structured Query Language (SQL)

Design, implement, evaluate and maintain relational database systems using SQL and Database Management Systems and explain the purpose of various SQL commands and operations.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦♦♦	Complete	1
Database Modelling Tools	♦♦♦♦♦	Complete	1
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	4
Database Normalisation	♦♦♦♦♦	Complete	2
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	2
Basic SQL	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	2
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	3
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	2
Learning Summary Report	♦♦♦♦♦	Complete	1
Database security	♦♦♦♦♦	Complete	1

3.4 Reflection

Analyse and critique the achievements of learning outcomes and justify meeting specified outcomes through providing relevant evidence and evaluating the quality of that evidence against given criteria.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦♦♦	Complete	1
Database Modelling Tools	♦♦♦♦♦	Complete	1
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	4
Database Normalisation	♦♦♦♦♦	Complete	2
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	2
Basic SQL	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	2
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	3
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	2
Learning Summary Report	♦♦♦♦♦	Complete	1
Database security	♦♦♦♦♦	Complete	1

3.5 Research and critical review

Conduct research on and critically evaluate tools and technologies used in contemporary business information systems to manage and analyse data.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦♦♦	Complete	1
Database Modelling Tools	♦♦♦♦♦	Complete	1
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	4
Database Normalisation	♦♦♦♦♦	Complete	2
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	2
Basic SQL	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	2
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	3
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	2
Learning Summary Report	♦♦♦♦♦	Complete	1
Database security	♦♦♦♦♦	Complete	1

4 Reflection on Data-driven Systems

Reflect on two data-driven information systems that you use in your daily life.

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

Outcome	Weight
Reflection	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

Outcome	Weight
Research and critical review	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

Date	**Author**	**Comment**
2025/03/11 22:48	Thi Ngo	Working On It
2025/03/13 15:00	Thi Ngo	Ready to Mark
2025/03/14 05:17	Mohammad Hossain	Very good.
2025/03/14 05:17	Mohammad Hossain	Complete
2025/03/17 11:15	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Reflection on Data-driven Systems

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/03/13 15:00

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

From this task, I have gained a deeper understanding of how data-driven information systems function, specifically through the case studies of Google Maps and YouTube. I learned how these platforms collect, store, and process large-scale structured and unstructured data, using relational (SQL) and NoSQL databases to manage location data, video content, and user interactions efficiently. Additionally, I explored the role of machine learning and artificial intelligence in improving recommendations, traffic predictions, and personalized user experiences. Beyond technical aspects, I also examined privacy and security concerns, such as data breaches, targeted advertising, and unauthorized tracking, and how companies implement encryption, access controls, and AI-driven moderation to mitigate these risks.

March 13, 2025

Reflection on Data-Driven Information Systems

Data-driven systems enhance daily life by collecting and processing structured and unstructured data to deliver personalized services. This report examines Google Maps and Youtube, focusing on their data usage, database structures, privacy concerns, and security measures.

Google Maps

1. Systems and Sort of Data

Google Maps is a consumer web mapping and navigation platform offering features real-time mapping, location tracking, and route planning. It collects various types of data, including GPS coordinates, traffic conditions, business locations, and user-generated content such as reviews and photos (Zheng, Yang & Li, 2021). The platform also provides satellite imagery, aerial photography, 360° interactive Street View, and real-time traffic conditions. As of 2020, over one billion people worldwide used Google Maps each month (Wikipedia, n.d.).

Google Maps aggregates its data from various sources including government agencies, third-party mapping providers, and user-contributed update to ensure accuracy and comprehensiveness:

Data Type	**Usage**
Geospatial Data	GPS coordinates, street maps, geographic boundaries.
Satellite Imagery	Bird's-eye views of landscapes and cities.
Street View Imagery	360-degree images from Street View cars and Trekker.
Real-Time User Data	GPS signals for live traffic updates and movement trends.

Traffic Information	Data from road sensors, mobile devices, and history to predict congestion.
Third-Party Data Providers	Business listings, reviews, and road network updates.
Public Data Sources	Government databases on roads, geography, and landmarks.
User-Generated Content	Reviews, ratings, and uploaded images.

As illustrated below, Google Maps collects geospatial data from satellites, aerial views, and crowdsourcing, processes it using algorithms, and integrates services for real-time navigation.

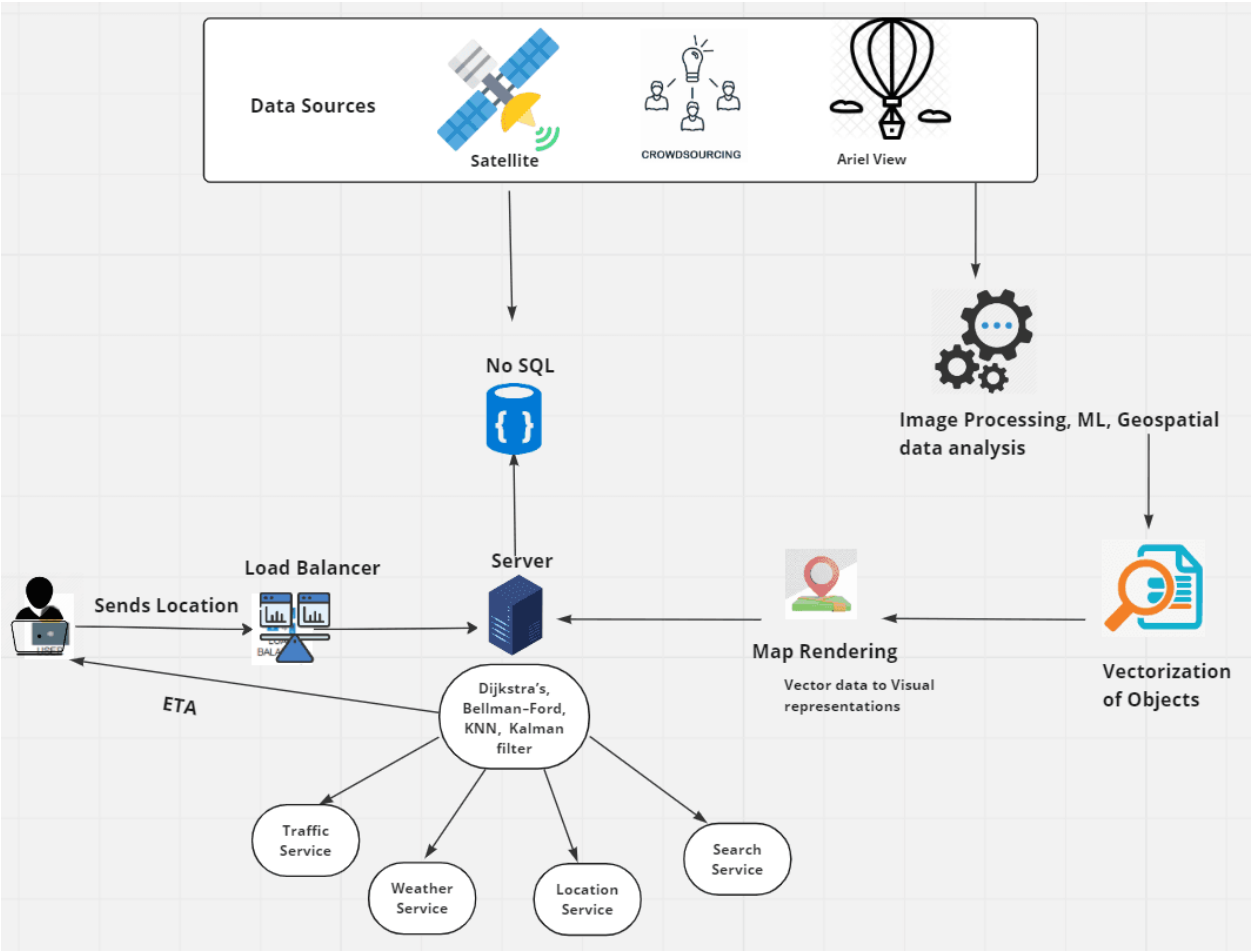


Figure 1: Google Maps System Design (Almabetter, 2023)

2. Data Usage and Availability

Google Maps utilizes geospatial databases to manage location-based data efficiently. Traffic updates are generated by analyzing mobile user data, road sensors, and historical trends (Zheng, Yang & Li, 2021). For instance, when users search for a restaurant, the platform retrieves results based on geolocation, preferences, and history.

By integrating real-time navigation and multi-modal transport data, Google Maps helps users identify efficient routes, avoid congestion, and plan seamless trips using public transport (Google, 2023). Without this data, users would face longer commutes, reduced access to traffic updates, and limited service recommendations. Additionally, the absence of congestion alerts could lead to higher fuel consumption and environmental impact (Ghafoorian et al., 2018), making Google Maps essential for urban mobility and sustainability.

3. Data Storage and Management

In terms of data storage and management, Google Maps relies on distributed databases, where data is stored across multiple servers worldwide to ensure scalability and real-time processing. It integrates SQL-based relational databases to manage structured location data and NoSQL databases such as Google Bigtable to handle large-scale, unstructured datasets like satellite imagery (Goodchild, 2020). To optimize query performance, Google Maps employs indexing and partitioning techniques, ensuring fast retrieval of geospatial data. For example, spatial indexing allows the system to quickly locate points of interest without scanning the entire database. This architecture enables high-speed processing of millions of user queries simultaneously.

4. Risks/concerns around security and privacy of the data

Privacy concerns arise due to continuous location tracking, which can expose sensitive information if mishandled. Security risks include unauthorized access to location history, targeted tracking, and data breaches (Google, 2023). Google mitigates these risks through end-to-end encryption, access controls, and privacy settings such as Incognito Mode, which prevents tracking of user locations. Despite these measures, concerns persist over third-party data sharing, particularly for advertising purposes. For example, businesses use Google’s data to target users based on their location history and browsing

behavior. Users must remain cautious when sharing personal location data to avoid potential misuse.

Youtube

1. Systems and Sort of Data

YouTube is a cloud-based video-sharing platform owned by Google, enabling users to upload, stream, and interact with video content globally. It operates on Google Cloud Platform (GCP) and utilizes a distributed content delivery network (CDN) to optimize video streaming. YouTube employs artificial intelligence (AI) and machine learning (ML) algorithms for content recommendations, moderation, and search ranking (Covington, Adams & Sargin, 2016).

Data Type	Usage
User-Generated Content	Videos uploaded by users, including metadata such as titles, descriptions, tags, and thumbnails. (Zhang et al., 2017).
Engagement Data	Likes, comments, shares, watch history, and subscriptions. (Covington, Adams & Sargin, 2016)
Behavioral Data	Watch duration, search history, click-through rates (CTR), playback preferences, and video interaction patterns (Covington, Adams & Sargin, 2016).
Advertising and Monetization Data	Ad impressions, revenue analytics, and user engagement with ads (Google, 2023)
Technical Data	Device type, browser, IP address, geolocation, internet speed, and playback resolution (Zhang et al., 2017).
Moderation and Policy Compliance Data	Copyright detection, community guidelines enforcement, and AI-driven content filtering (Google, 2023).

YouTube's system architecture processes video requests through web servers, load balancers, and API servers, ensuring efficient data handling. Video content is stored,

transcoded, and distributed via a Content Delivery Network (CDN), while metadata caching optimizes search functionality, recommendations, and streaming performance.

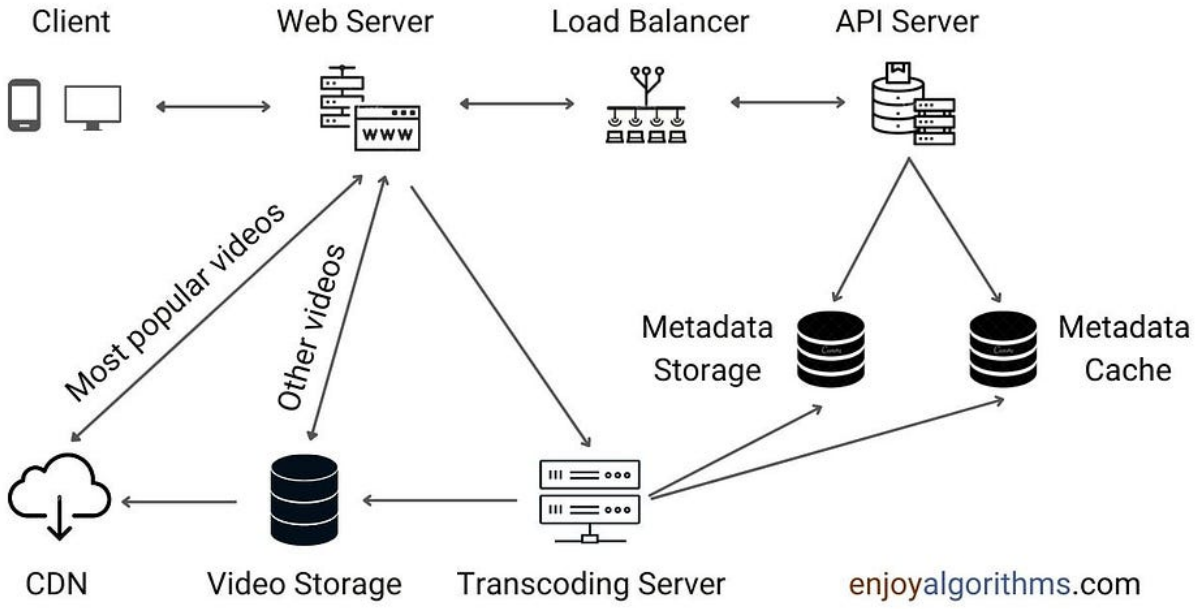


Figure 2: Youtube System Design (EnjoyAlgorithm, 2025)

2. Data Usage and Availability

YouTube leverages data to enhance content recommendations, personalize user experiences, and optimize video streaming quality. Through machine learning algorithms, it analyzes watch history, search patterns, and engagement metrics to suggest relevant videos (Covington, Adams & Sargin, 2016). The real-time adaptive streaming feature ensures smooth playback by adjusting video quality based on network conditions and device capabilities. Additionally, YouTube's data-driven approach supports content creators by providing audience insights, helping them optimize content strategies. Availability is ensured through globally distributed content delivery networks (CDNs), allowing seamless access to videos from any location. This enables high-performance, low-latency video delivery, ensuring millions of concurrent users experience minimal buffering.

3. Data Storage and Management

YouTube operates on a highly scalable cloud-based storage system, utilizing Google Cloud Storage (GCS) to handle large volumes of video files. Structured data, such as user accounts, metadata, and engagement statistics, is managed using SQL-based relational databases, while NoSQL databases like Bigtable and BigQuery store vast amounts of unstructured video data (Zhang et al., 2017).

To optimize storage and retrieval, YouTube employs data compression, indexing, and caching techniques. Videos are stored in multiple resolutions and formats to support adaptive streaming, ensuring efficient bandwidth usage and improved playback performance. Content Delivery Networks (CDNs) distribute videos globally, reducing latency and enabling seamless access. Additionally, machine learning models manage content recommendations, spam detection, and copyright protection, improving data efficiency and security (Google, 2023).

4. Risks/concerns around security and privacy of the data

YouTube faces significant security and privacy risks, including unauthorized data access, misinformation, AI-generated bias, and targeted advertising concerns. User data, including watch history, search queries, and personal preferences, is collected for personalization and ad targeting, raising privacy concerns (Google, 2023). Additionally, data breaches and account hacking pose security threats. Misinformation and harmful content also challenge content moderation efforts. YouTube mitigates risks through end-to-end encryption, privacy controls, AI-based moderation, and two-factor authentication, but ethical concerns regarding data transparency and algorithmic biases persist (Statista, 2024).

References

AlmaBetter, 2025. *System design of Google Maps*. [online] Available at:

https://www.almabetter.com/bytes/articles/system-design-of-google-maps

[Accessed 11 March 2025].

Covington, P., Adams, J. & Sargin, E., 2016. *Deep neural networks for YouTube recommendations*. *Proceedings of the 10th ACM Conference on Recommender Systems*, pp.191-198.

EnjoyAlgorithms, 2025. *Design YouTube: System design interview question*. [online] Available at: https://www.enjoyalgorithms.com/blog/design-youtube-system-design-interview-question [Accessed 11 March 2025].

Ghafoorian, M., Kámán, Z., Jermyn, I. & Wegner, J.D., 2018. *Traffic flow prediction with artificial intelligence techniques: A comprehensive review*. *Transportation Research Part C: Emerging Technologies*, 96, pp.225-243.

Goodchild, M.F., 2020. *Spatial thinking and data-driven decision-making in a smart world*. *Journal of Spatial Information Science*, 2020(21), pp.1-19.

Google, 2023. *Privacy settings in Google Maps*. [online] Available at: https://www.google.com/privacy [Accessed 11 March 2025].

Statista, 2024. *YouTube users worldwide*. [online] Available at: https://www.statista.com/statistics [Accessed 11 March 2025].

Wikipedia, n.d.. *Google Maps*. [online] Available at: https://en.wikipedia.org/wiki/Google_Maps [Accessed 11 March 2025].

Zheng, Y., Yang, Z. & Li, Z., 2021. *Urban computing: Enabling data-driven solutions for smart cities*. *IEEE Internet of Things Journal*, 8(5), pp.3358-3372.

Zhang, Y., Liu, X., Sakamoto, T. & Lin, H., 2017. *Efficient cloud-based video storage and retrieval system for large-scale multimedia applications*. *Multimedia Tools and Applications*, 76(21), pp.22309-22326.

5 Install MySQL

Install the MySQL community server and workbench.

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

Outcome	Weight
Reflection	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

Outcome	Weight
Research and critical review	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

Date	Author	Comment
2025/03/11 22:48	Thi Ngo	Working On It
2025/03/11 22:58	Thi Ngo	Ready to Mark
2025/03/11 22:58	Thi Ngo	I have submitted the required task, I highly appreciate for your feedback.
2025/03/13 14:02	Thi Ngo	Not Started
2025/03/13 14:05	Thi Ngo	Ready to Mark
2025/03/14 14:00	Mohammad Hossain	It's all good.
2025/03/14 14:00	Mohammad Hossain	Complete
2025/03/14 14:04	Mohammad Hossain	Complete
2025/03/17 11:15	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Install MySQL

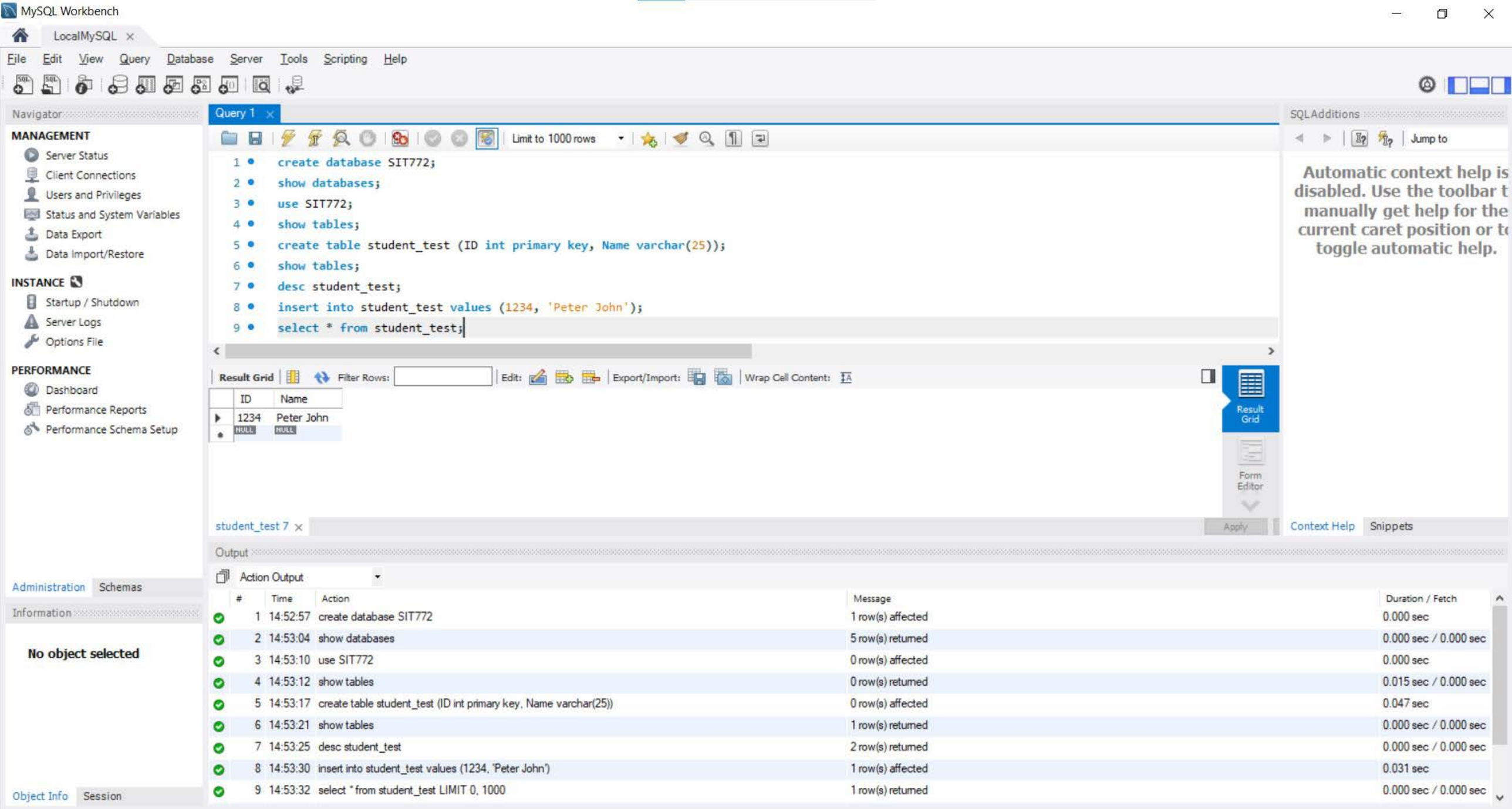
Submitted By:
Thi Thu Suong Ngo
s224757434
2025/03/13 14:05

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

From this task, I learned how to set up and use MySQL Workbench to interact with a database. I practiced executing basic SQL commands such as creating a database, defining tables, inserting data, and retrieving records using SELECT. Additionally, I gained an understanding of primary keys and how they enforce uniqueness in a table. This exercise helped me develop foundational skills in database management and querying, which are essential for handling structured data efficiently.

March 13, 2025


MySQL Workbench
LocalMySQL
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
1 create database SIT772;
2 show databases;
3 use SIT772;
4 show tables;
5 create table student_test (ID int primary key, Name varchar(25));
6 show tables;
7 desc student_test;
8 insert into student_test values (1234, 'Peter John');
9 select * from student_test;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
ID Name
1234 Peter John
NULL NULL
Result Grid
Form Editor
Apply
student_test 7
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 14:52:57 create database SIT772 1 row(s) affected 0.000 sec
2 14:53:04 show databases 5 row(s) returned 0.000 sec / 0.000 sec
3 14:53:10 use SIT772 0 row(s) affected 0.000 sec
4 14:53:12 show tables 0 row(s) returned 0.015 sec / 0.000 sec
5 14:53:17 create table student_test (ID int primary key, Name varchar(25)) 0 row(s) affected 0.047 sec
6 14:53:21 show tables 1 row(s) returned 0.000 sec / 0.000 sec
7 14:53:25 desc student_test 2 row(s) returned 0.000 sec / 0.000 sec
8 14:53:30 insert into student_test values (1234, 'Peter John') 1 row(s) affected 0.031 sec
9 14:53:32 select * from student_test LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec

6 Database Modelling Tools

Fundamental concepts of relational database modelling and tools for database modelling

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

Outcome	Weight
Reflection	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

Outcome	Weight
Research and critical review	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such

as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

Date	**Author**	**Comment**
2025/03/14 13:19	Thi Ngo	Working On It
2025/03/19 19:02	Thi Ngo	Ready to Mark
2025/03/21 14:39	Mohammad Hossain	Excellent work.
2025/03/21 14:39	Mohammad Hossain	Complete
2025/03/30 15:58	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Database Modelling Tools

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/03/19 19:02

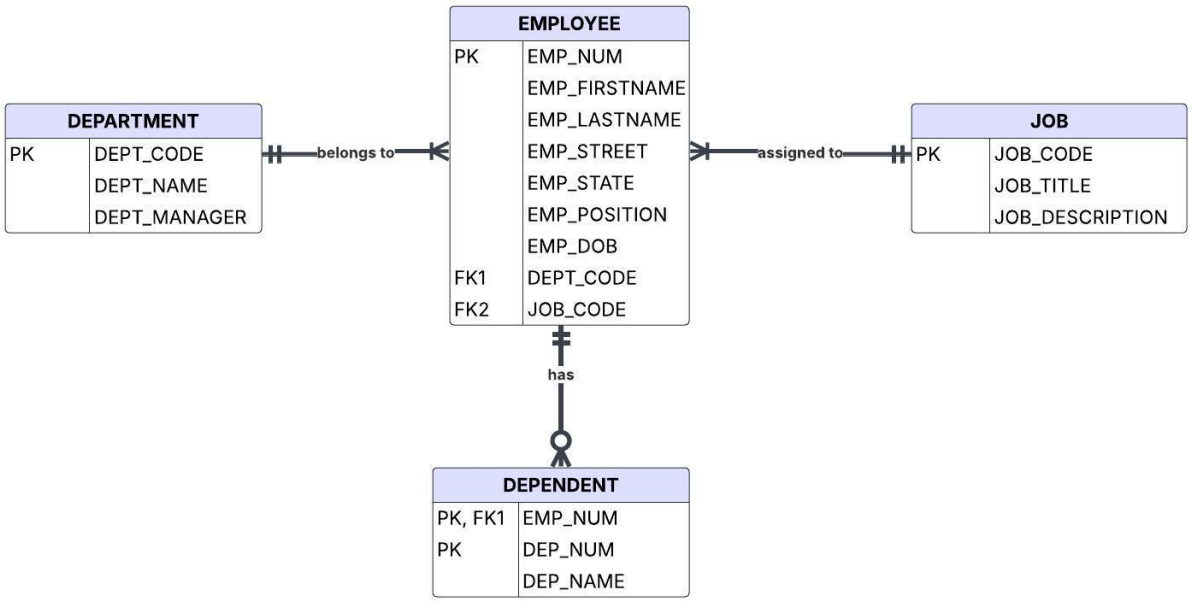
Tutor:
Mohammad belayet Hossain

Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

From this task, I have gained a deeper understanding of relational database modeling and how to represent entities, relationships, and constraints using an Entity-Relationship Diagram (ERD). I learned how to use Lucidchart to design ERDs effectively, including defining primary keys, foreign keys, and composite keys. I also explored cardinality and the nature of relationships between entities, such as the 1:N relationship between EMPLOYEE and DEPENDENT. Additionally, I practiced creating a context diagram, which provides a high-level overview of a system's data flow. This task has strengthened my ability to design well-structured databases and apply fundamental database principles to real-world scenarios.

March 19, 2025

Task 3 – Drawing the following ERD using LucidChart



Task 4 – Answering questions based on drawn ERD

a. How many Entities are there and what are they?

There are four entities in the ERD including:

1. EMPLOYEE
2. DEPARTMENT
3. JOB
4. DEPENDENT

b. For the EMPLOYEE Entity: (i) List all attributes; (ii) What is the Primary Key and how do you know that? and (iii) Are there any Foreign Key(s)? If yes, what are they and what are they referring to?

For the EMPLOYEE Entity:

(i) All attributes includes:

- EMP_NUM (Primary Key)
- EMP_FIRSTNAME
- EMP_LASTNAME
- EMP_STREET

- EMP_STATE
- EMP_POSITION
- EMP_DOB
- DEPT_CODE (Foreign Key)
- JOB_CODE (Foreign Key)

(ii) The Primary Key of the EMPLOYEE entity is EMP_NUM. It is marked with PK in the ERD, indicating that it uniquely identifies each employee.

(iii) There are two Foreign Keys:

- DEPT_CODE → References the DEPARTMENT entity (DEPT_CODE).
- JOB_CODE → References the JOB entity (JOB_CODE).

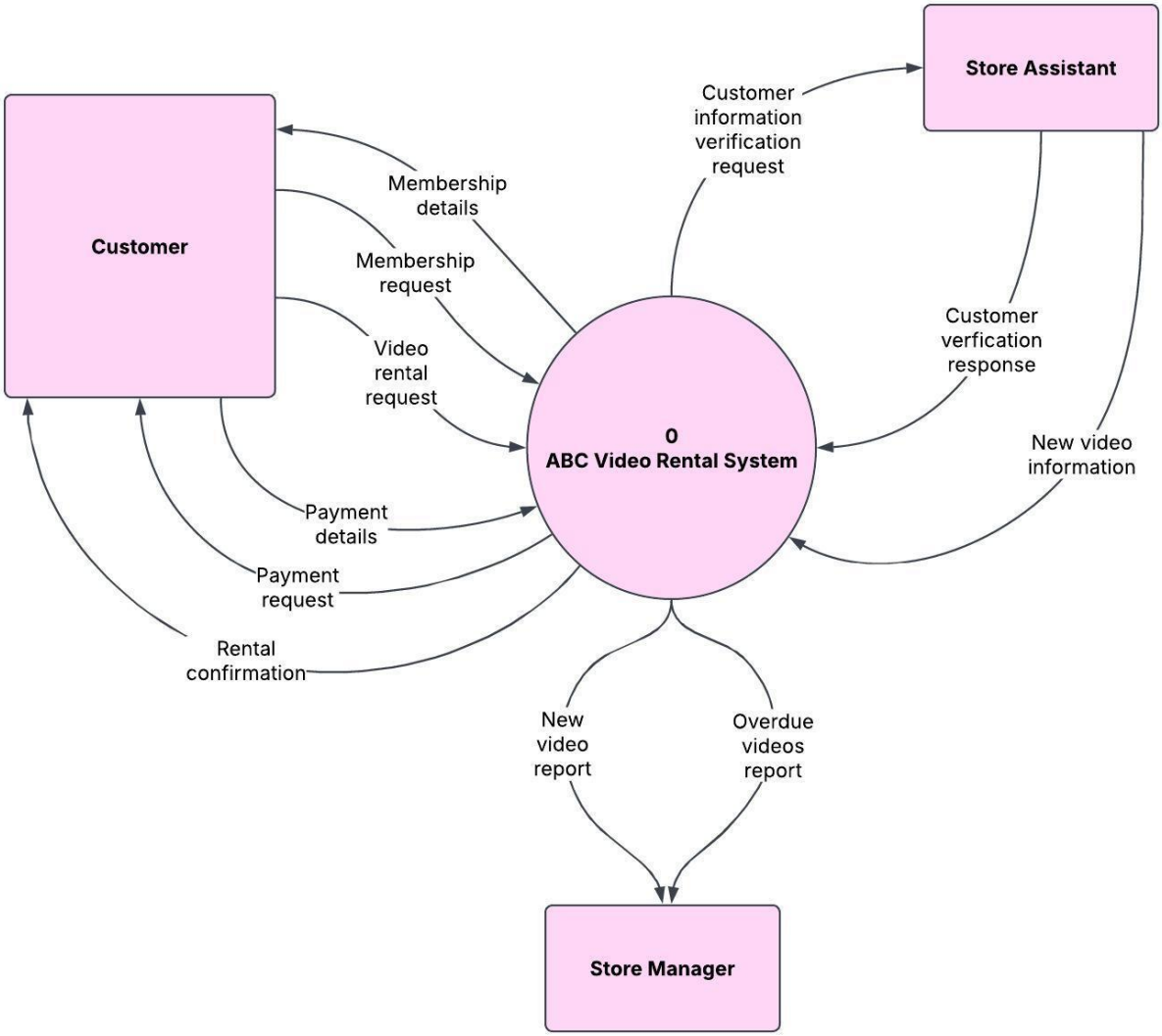
c. Explain the relationship between EMPLOYEE and DEPENDENT.

- **Cardinality:**
  - An EMPLOYEE can have multiple DEPENDENTs (1:N relationship).
  - A DEPENDENT must be associated with an EMPLOYEE.
- **Optional/ Mandatory:**
  - The relationship is mandatory for DEPENDENT (every dependent must be linked to an employee).
  - The relationship is optional for EMPLOYEE (not every employee must have dependents).
- **Maintaining the relationship**: To ensure that each dependent is associated with a valid employee
  - The Primary Key of DEPENDENT consists of EMP_NUM and DEP_NUM.
  - EMP_NUM in DEPENDENT is a Foreign Key referencing EMP_NUM in EMPLOYEE.

d. What is the Primary Key of DEPENDENT? What type of key is it?

- The Primary Key of DEPENDENT is a Composite Key consisting of:
  - EMP_NUM (Foreign Key from EMPLOYEE)
  - DEP_NUM (Unique identifier for each dependent of an employee)
- Type of Key: Composite Primary Key (as it consists of two attributes).

Task 5 – Drawing Context Diagram for a Video Rental System using Lucid Chart



7 Entity Relationship Diagram

Analyse data requirements of a given system and draw an ERD

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

Outcome	Weight
Reflection	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

Date	Author	Comment
2025/03/25 10:49	Thi Ngo	Working On It
2025/03/26 22:37	Thi Ngo	Ready to Mark
2025/03/27 20:31	Thi Ngo	Not Started
2025/03/27 20:31	Thi Ngo	Ready to Mark
2025/03/29 00:36	Mohammad Hossain	Good. However, you could consider 6 entities (such as PERSON, PATIENT, PRESCRIPTION, DOCTOR, DRUG, DRUG_COMPANY) for improving the ERD.
2025/03/29 00:36	Mohammad Hossain	Complete
2025/03/30 16:03	Thi Ngo	Many thanks Mr. Mohammad for your feedback and I will consider 6 entities for further optimizing the ERD structure. I agree that introducing an additional general entity such as PERSON could indeed enhance the ERD design, as it could generalize common attributes shared by PATIENT and DOCTOR (like name, contact details, and address). Separating the entities into PERSON, PATIENT, and DOCTOR would help avoid redundancy and better reflect the real-world hierarchy. Thank you again for the constructive feedback!
2025/03/30 22:18	Mohammad Hossain	No worries.

Deakin University

Database Fundamentals

OnTrack Submission

Entity Relationship Diagram

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/03/27 20:31

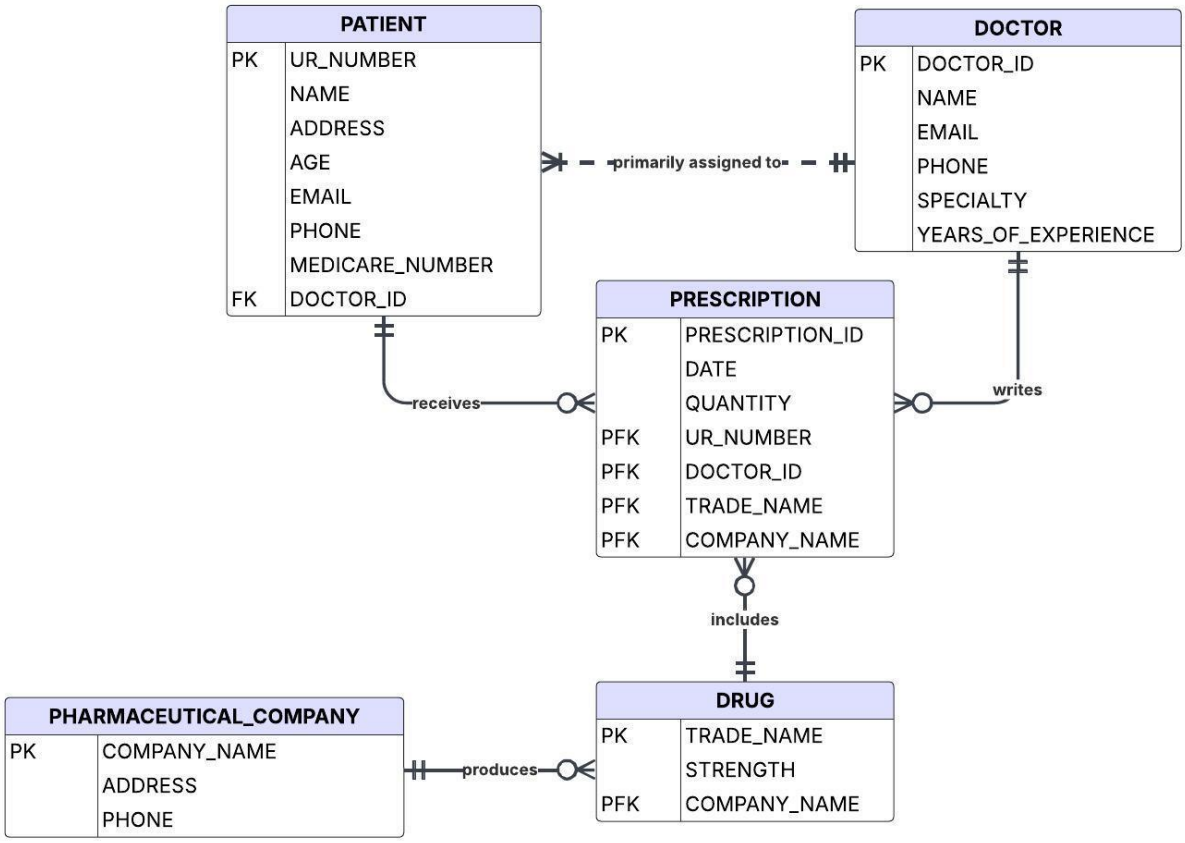
Tutor:
Mohammad belayet Hossain

Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through this task, I learned how to analyze a real-world business scenario and translate it into a structured Entity Relationship Diagram (ERD) using Crow's Foot notation. I developed a deeper understanding of how to identify entities, attributes, primary and foreign keys, and how to accurately model complex relationships, including many-to-many associations using associative entities. Additionally, I learned the importance of enforcing data integrity through constraints like referential integrity and cascade deletes. This exercise has strengthened my ability to design clear, normalized database models that reflect practical data requirements.

March 27, 2025

Below is the Entity Relationship Diagram (ERD) of the proposed Barwon Health Prescription System based on Crow's Foot notation:



1. **All Entities, their Attributes, Constraints, and Relationships with Cardinalities**

PATIENT

- PK: UR_NUMBER
- Attributes: NAME, ADDRESS, AGE, EMAIL, PHONE, MEDICARE_NUMBER
- FK: DOCTOR_ID (primary doctor)
- Constraints: Every patient must have a primary doctor, UR_NUMBER is unique

DOCTOR

- PK: DOCTOR_ID
- Attributes: NAME, EMAIL, PHONE, SPECIALTY, YEARS_OF_EXPERIENCE
- Constraints: A doctor must be assigned to at least one patient

PHARMACEUTICAL_COMPANY

- PK: COMPANY_NAME
- Attributes: ADDRESS, PHONE
- Constraints: Cascade delete – if a company is deleted, so are its drugs

DRUG

- PK: TRADE_NAME
- Attributes: STRENGTH
- PFK: COMPANY_NAME
- Constraints: A drug is always linked to one company

PRESCRIPTION

- PK: PRESCRIPTION_ID
- Attributes: DATE, QUANTITY
- Composite FK: UR_NUMBER, DOCTOR_ID, TRADE_NAME, COMPANY_NAME
- Constraints: Connects PATIENT, DOCTOR, and DRUG

2. **Identify the Primary Key (PK) in all entities.**

Entity	Primary Key
PATIENT	UR_NUMBER
DOCTOR	DOCTOR_ID
DRUG	TRADE_NAME (with COMPANY_NAME as PFK)
PHARMACEUTICAL_COMPANY	COMPANY_NAME
PRESCRIPTION	PRESCRIPTION_ID

3. **Implement and label all relationships using Primary/Foreign Key (PK/FK) or Associative Entities**

- **PATIENT — DOCTOR**

- Relationship: *"Primarily assigned to"*
- Type: One or many (1,N) patient to one and only one doctor (1,1)
- Implementation: PATIENT has FK DOCTOR_ID referencing DOCTOR
- PATIENT is a strong entity with its own PK: UR_NUMBER and PATIENT's identity does not depend on DOCTOR.

- **PRESCRIPTION as Associative Entity**:

- Connects PATIENT (UR_NUMBER), DOCTOR (DOCTOR_ID), and DRUG (TRADE_NAME, COMPANY_NAME)
- Also contains its own PK: PRESCRIPTION_ID
- **DOCTOR — PRESCRIPTION:** One and only one doctor (1,1) writes zero or many prescriptions (0,N)
- **PATIENT — PRESCRIPTION:** One and only one patient (1,1) receives zero or many prescriptions (0,N)
- **DRUG — PRESCRIPTION:** One and only one drug (1,1) can be part of zero or many prescriptions (0,N)
- **PHARMACEUTICAL_COMPANY — DRUG:** One and only one Pharmaceutial_Company (1,1) produces zero or many drugs (0,N) with cascade delete

4. **Consider using specialization/generalization relationships using supertype and subtype entities with disjoint/overlapping and partial/total completeness constraints if possible.**

   Currently, due to distinct attributes and clear roles, treating Patient and Doctor as separate entities is more appropriate. No specialization/generalization is needed in this scenario.

5. **Justifications for entities, attributes, constraints, and relationships/cardinalities you considered**

- **Entity Choices:**

- PATIENT and DOCTOR are clearly required to model core hospital interactions.
- PRESCRIPTION is an associative entity capturing the many-to-many relationship between doctors and patients, including necessary attributes like date and quantity.
- DRUG and PHARMACEUTICAL_COMPANY represent the suppliers and products, essential for the prescription system.

- **Relationships:**

- One or many patient primarily assign to one and only one doctor. There is a weak relationship as the primary key of Doctor_ID is not included as PK in Patient, and also in real business case, if patient holds Doctor_ID as PK, it is impossible since if the doctor resigns, the patient data still continuously available and the value of Doctor can be null → Hence a foreign key in PATIENT.
- A doctor can write zero or many prescriptions, there is strong relationship since the Doctor's PK (DOCTOR_ID) is included in Prescription as part of its PK (DOCTOR_ID as PFK).
- Each patient receives zero or many . There is strong relationship as Patient's PK (UR_NUMBER) is part of the Prescription's composite PK (UR_NUMBER as PFK).
- A drug belongs to one pharmaceutical company. Cascade delete on COMPANY is required to maintain referential integrity. There is strong relationship as **COMPANY_NAME** is part of the **DRUG** primary key. Thus, the removal of a pharmaceutical company necessitates a **cascade delete** of all its associated drugs to maintain referential integrity and prevent orphaned drug records."

Summary table of Relationship

Entity 1	Relationship	Entity 2	Cardinality
PATIENT	Primarily assigned_to	DOCTOR	(1,N) – (1,1)
PATIENT	receives	PRESCRIPTION	(1,1) – (0,N)
DOCTOR	writes	PRESCRIPTION	(1,1) – (0,N)
PRESCRIPTION	includes	DRUG	(0,N) – (1,1)
PHARMACEUTICAL_COMPANY	produces	DRUG	(1,1) – (0,N)

8 Database Normalisation

Draw Dependency diagram and Normalise a given database table

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

Outcome	Weight
Reflection	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

Date	**Author**	**Comment**
2025/03/27 14:22	Thi Ngo	Working On It
2025/04/03 20:06	Thi Ngo	Ready to Mark
2025/04/05 07:51	Mohammad Hossain	The Manuscript_NUM and Author_NUM are the PK while Manuscript_NUM has partial relationship with submission_date, Editor_num, and Feedback_recommended. The Author_NUM has partial relationship with Author_Name and Author_Institute.There is a transitive dependency between Author_Institution and Institution_Country as well as between Editor_num and Editor_Name. Please draw the dependency diagrams as well as ERD based on the above mentioned relationship.For 2NF, there should be 2 tables and for 3NF, there should be 4 tables.
2025/04/05 07:52	Mohammad Hossain	As there is no common non-key attribute, you can remove Journal_entry table/entity from both 2NF and 3NF. As a result, for ERD there should be 4 entities.
2025/04/05 07:52	Mohammad Hossain	Fix and Resubmit
2025/04/06 17:32	Thi Ngo	Ready to Mark
2025/04/06 17:36	Thi Ngo	Thank you for your feedback. I have now adjusted the model from five entities to four by removing the associative entity Journal_Entry, since it had no non-key attributes and served only to represent the many-to-many relationship between AUTHOR and MANUSCRIPT. In the updated ERD, the four remaining entities—AUTHOR, MANUSCRIPT, EDITOR, and INSTITUTION—are in 3NF, with partial dependencies eliminated in 2NF and transitive dependencies (such as AUTHOR_INSTITUTION undefined INSTITUTION_COUNTRY and EDITOR_NUM undefined EDITOR_NAME) properly resolved.
2025/04/13 16:28	Mohammad Hossain	Very good. Thanks.
2025/04/13 16:28	Mohammad Hossain	Complete
2025/04/14 16:37	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Database Normalisation

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/06 17:32

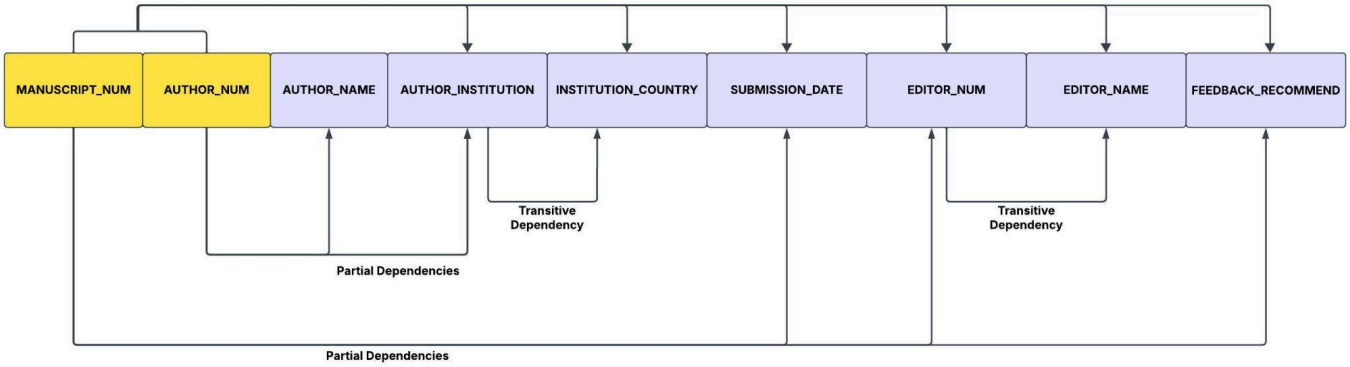
Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through this task, I learned how to apply the principles of database normalization to transform an initial, unorganized dataset into a well-structured design. Specifically, I gained experience in identifying and removing partial dependencies in Second Normal Form (2NF), which involves breaking down tables to ensure that all attributes depend on the full primary key. I also learned how to address transitive dependencies in Third Normal Form (3NF) by moving non-key attributes that depend on other non-key attributes into separate tables. Finally, I developed an understanding of how to construct Entity Relationship Diagrams (ERD) using Crow's Foot notation to visually represent the relationships between normalized tables. This task enhanced my ability to design efficient, scalable databases with clear relationships and minimal redundancy.

April 6, 2025

First Normal Form



Second Normal Form

Table name: AUTHOR

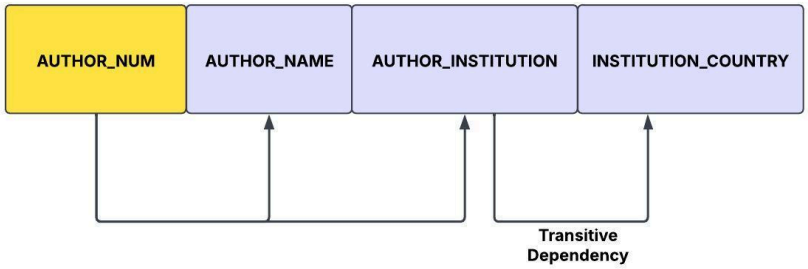
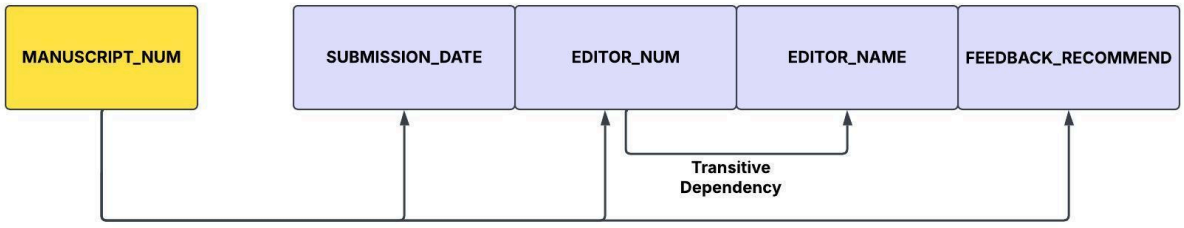
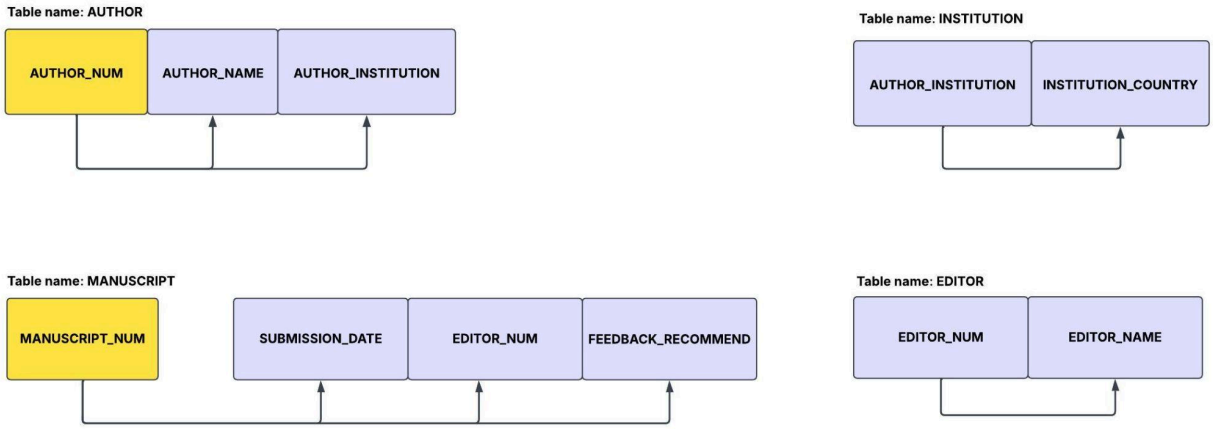


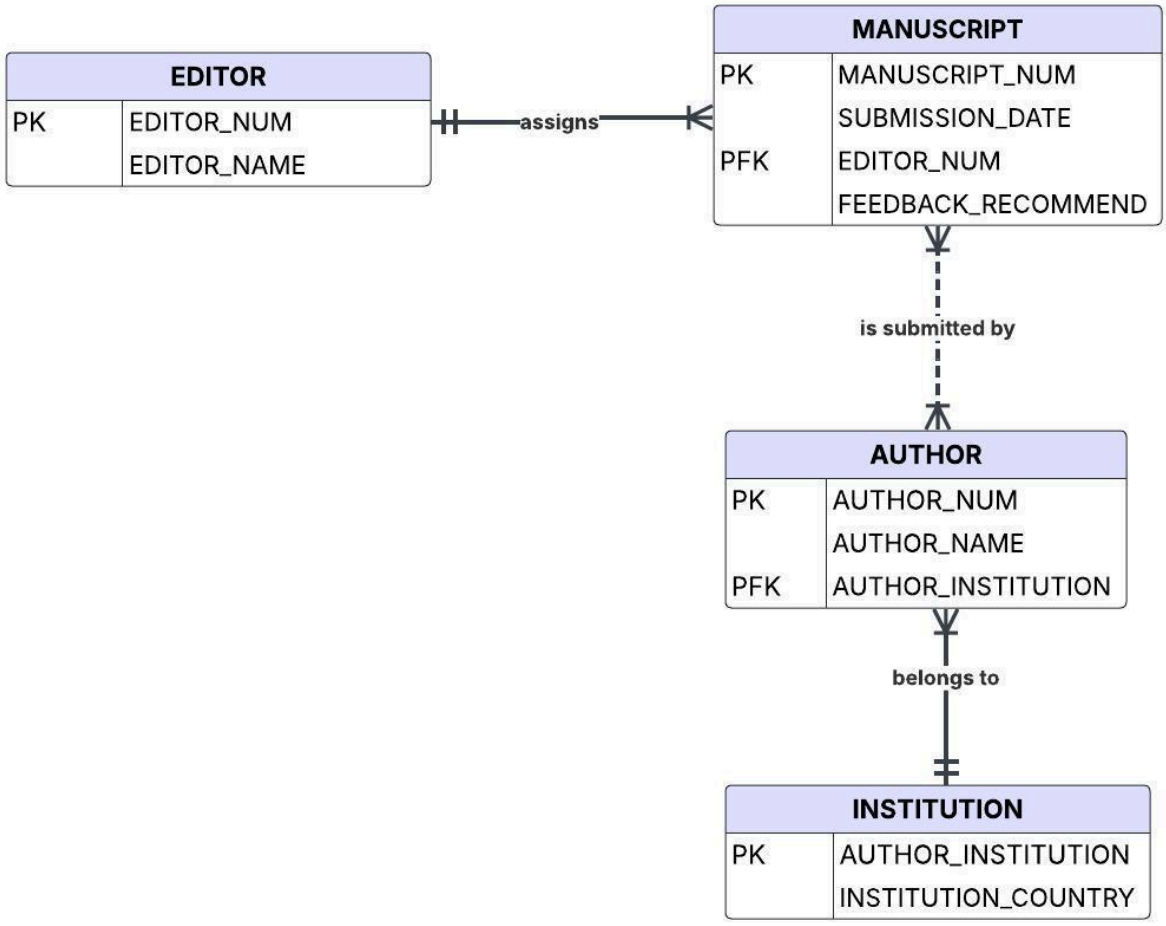
Table name: MANUSCRIPT



Third Normal Form



Entity Relationship Diagram (ERD) of the database using Crow Foot notations



This Entity-Relationship Diagram (ERD) models a manuscript submission and review system. It includes four key entities: **EDITOR**, **MANUSCRIPT**, **AUTHOR**, and **INSTITUTION**, along with their attributes, constraints, relationships, and cardinalities.

EDITOR:

- PK: `EDITOR_NUM`
- Attributes: `EDITOR_NAME`.

MANUSCRIPT:

- PK: `MANUSCRIPT_NUM` with `EDITOR_NUM` (PFK)
- Attributes: `SUBMISSION_DATE`, `FEEDBACK_RECOMMEND`

AUTHOR:

- PK: `AUTHOR_NUM with AUTHOR_INSTITUTION` (PFK)
- Attribute: `AUTHOR_NAME`

INSTITUTION:

- PK: `AUTHOR_INSTITUTION`
- Attribute: `INSTITUTION_COUNTRY`.

2. Relationships Explanation

- **EDITOR - MANUSCRIPT**

- Relationship: “assigns”
- Type: One and only one editor (1,1) to one or many manuscripts (1,N)
- Cardinality: EDITOR (1,1) → assigns → MANUSCRIPT (1,N)
- Justification: Each manuscript must be assigned to exactly one editor upon submission. Each editor can be responsible for multiple manuscripts. There is a strong relationship as EDITOR_NUM is part of MANUSCRIPT primary foreign key.

- **MANUSCRIPT - AUTHOR**

- Relationship: *“is written by”*
- Type: One or many (1, N) manuscript to one or many author (1, N)
- Cardinality: MANUSCRIPT (1,N) → is submitted by → AUTHOR (1,N)
- Justification: Each MANUSCRIPT can be written by one or more AUTHORS. Each AUTHOR can write one or more MANUSCRIPTS. MANUSCRIPT is a strong entity with its own PK: MANUSCRIPT_NUM and MANUSCRIPT’s identity does not depend on AUTHOR.

- **AUTHOR - INSTITUTION**

- Relationship: *“belongs to”*
- Type: One or many authors (1,N) to one and only one institution (1,1)
- Cardinality: AUTHOR (1,N) → has → INSTITUTION (1,1)
- Justification: Each author must be affiliated with exactly one institution. Each institution can have multiple authors. There is a strong relationship as AUTHOR_INSTITUTION is part of AUTHOR primary foreign key.

9 Miniproject Part-1 - Database Design and Normalisation

Analyse data requirements of a company, design and normalised a database to meet their needs.

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

Outcome	Weight
Reflection	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

Date	Author	Comment
2025/03/27 14:24	Thi Ngo	Working On It
2025/04/02 20:37	Thi Ngo	Not Started
2025/04/03 20:06	Thi Ngo	Working On It
2025/04/07 16:34	Thi Ngo	I am working on a complex project for Trip.com database so I would like to spend extra time to make sure I ensure the quality outcome of the project. I have headache currently so I would like to have extension to ensure for my wellbeing.
2025/04/16 13:56	Thi Ngo	Ready to Mark
2025/04/16 14:00	Thi Ngo	Dear Dr. Mohammad, I am writing to update that I have changed the company for my mini project from Trip.com to Coursera. I found that Coursera's structure allowed for a more thorough and meaningful data model. Highly appreciate if this change is alright for my side.Best Regards,Suong Ngo.
2025/04/24 19:29	Mohammad Hossain	What is PFK in your ERD? Please fix and resubmit.
2025/04/24 19:29	Mohammad Hossain	Fix and Resubmit
2025/04/27 17:43	Thi Ngo	Dear Tutor, I feel unwell and I look forward to having your support to allow me to extend the deadline until 30th April so that I can ensure the task outcome better.
2025/04/27 23:49	Thi Ngo	Working On It
2025/04/27 23:51	Thi Ngo	Ready to Mark
2025/04/27 23:51	Mohammad Hossain	Time Exceeded
2025/04/27 23:51	Thi Ngo	Dear Dr. Mohammad,I have fixed the notation issue by replacing 'PFK' with 'PK' and 'FK' as per the standard ERD conventions.I have also rechecked the relationships and entity structures. Please find the corrected ERD attached. Thank you for your feedback and guidance.Best Regards,Suong Ngo.
2025/04/27 23:55	Thi Ngo	Dear Tutor, I feel unwell during the holiday and tried my best to complete today. I missed for feedback for correction and have to complete within the proposed timeline. I look forward to having your support for extension for this task for your review.
2025/04/30 09:57	Thi Ngo	Dear Dr. Mohammad,I look forward to having your urgent support as I have resubmitted my task on 27th of April, but I think because it pass the due date after having your feedback and it only allows me for one day to fix so I have missed it. I look forward to having your consideration to mark this task as it shows I have passed the due dates for 5 days while I have resubmitted the task.Thank you very much for your kind support.Best Regards,
2025/04/30 09:57	Thi Ngo	Suong Ngo
2025/04/30 09:58	Thi Ngo	image comment
2025/05/04 16:42	Thi Ngo	Dear Dr. Mohammand, I need for your urgent support for this task as I have resubmitted one week ago, but it keeps showing on the screen I have past the due dates in which the due dates keep increasing date by date. I need to have your urgent review and support on this task so that I can perform Mini Project Part 2.Thank you very much for your support.Best Regards,Suong Ngo.

Deakin University

Database Fundamentals

OnTrack Submission

Miniproject Part-1 - Database Design and Normalisation

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/27 23:51

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through completing this task, I gained a deeper understanding of how to analyze and model real-world business processes using Entity-Relationship Diagrams (ERDs). I learned how to identify strong and weak entities, establish appropriate primary and foreign keys, and define relationship cardinalities and constraints. Additionally, by applying normalization rules up to Third Normal Form (3NF), I ensured that the database design is efficient, consistent, and free of redundancy. This task has enhanced my skills in database design and logical data modeling in a structured, professional context.

April 27, 2025

1) Organization Overview and Business Processes

- **Company Name:** Coursera
- **Website:** https://www.coursera.org
- **Main Business:** Coursera is an online education platform that collaborates with universities, colleges, and industry partners to provide courses, professional certificates, specializations, and degree programs to learners worldwide. It enables accessible, flexible, and affordable learning across a broad range of subjects.
- **Business Description:** Coursera serves millions of learners globally by providing access to high-quality educational content through the internet. It partners with over 300 leading universities and companies including Stanford, Google, and IBM. Learners can browse a wide range of courses, enroll, study at their own pace, complete assessments, and earn credentials.
- **Core Business Processes, Functions, and Activities:**
  - ❖ **For Learners (Users):**
    - Account Registration: Users sign up with personal information (name, email, etc.).
    - Course Browsing: Users explore available courses by subject, institution, or skill level.
    - Enrollment: Users can enroll in courses for free or pay for certification.
    - Learning Progress: Users access videos, readings, quizzes, and assignments.
    - Assessments: Users complete graded assignments and peer-reviewed tasks.
    - Certification: Upon successful completion, a certificate is generated.
    - Subscription/Payment: Some users pay per course or subscribe monthly.
  - ❖ **For Instructors and Institutions:**
    - Course Creation: Instructors and partner institutions create and manage course content (lectures, quizzes, readings, projects).
    - Grading & Feedback: Instructors and systems evaluate assignments.
    - Analytics: Institutions review learner engagement and performance statistics.

- **Scenarios and Use Cases**

Scenario	Actors	Precondition	Steps	Use Case Outcome & Business Rules
1. User Enrolls in a Course	User, System	User has an account	1. User searches a course 2. Clicks "Enroll" 3. Payment (if required) 4. Enrollment created 5. Access granted	The Enrollment entity is created linking the User and the Course. Business rule: A user can only enroll in a course once at a time.
2. Learner Completes Assignments and Gets Graded	User, Instructor, System	User is enrolled in a course	1. User opens an assignment. 2. User uploads their work before the due date. 3. System auto-grades or forwards to peer/instructor for marking. 4. Grade is stored, feedback provided.	A Submission entity is created with a timestamp and score. Business rule: One submission per user per assignment.
3. Course Completion and Certificate Generation	System, User	All course requirements met (assignments, quizzes, etc.)	1. System checks course completion status. 2. If complete, a certificate is automatically generated. 3. Certificate is linked to the user's profile.	A Certificate record is created, linked to the Enrollment. Business rule: A certificate is only issued if a course is completed.
4. Instructor Creates and Manages a Course	Instructor, System	Instructor is approved by the institution	1. Instructor logs in and starts a new course. 2. They add video lectures, quizzes, and assignments. 3. Course is saved as a draft or published.	A Course is created with associated Assignment records. Business rule: A course must be linked to at least one instructor and one institution.

5. Institution Partners with Coursera	Coursera Admin, Institution	Partnership agreement is signed	1. Institution profile is created. 2. Institution invites instructors and uploads their first course. 3. Courses appear on Coursera with institution branding.	An Institution entity is created, linked to Instructors and Courses. Business rule: An institution can have many instructors and courses.
6. User Subscribes to Coursera Plus	User, Payment System	User has account	1. User chooses subscription plan. 2. Payment is processed. 3. Subscription record is activated. 4. User can now access eligible paid courses without paying again.	A Subscription entity may be created with payment history. Business rule: Subscription can expire or auto-renew.

2) All Entities with Attributes and Primary Keys

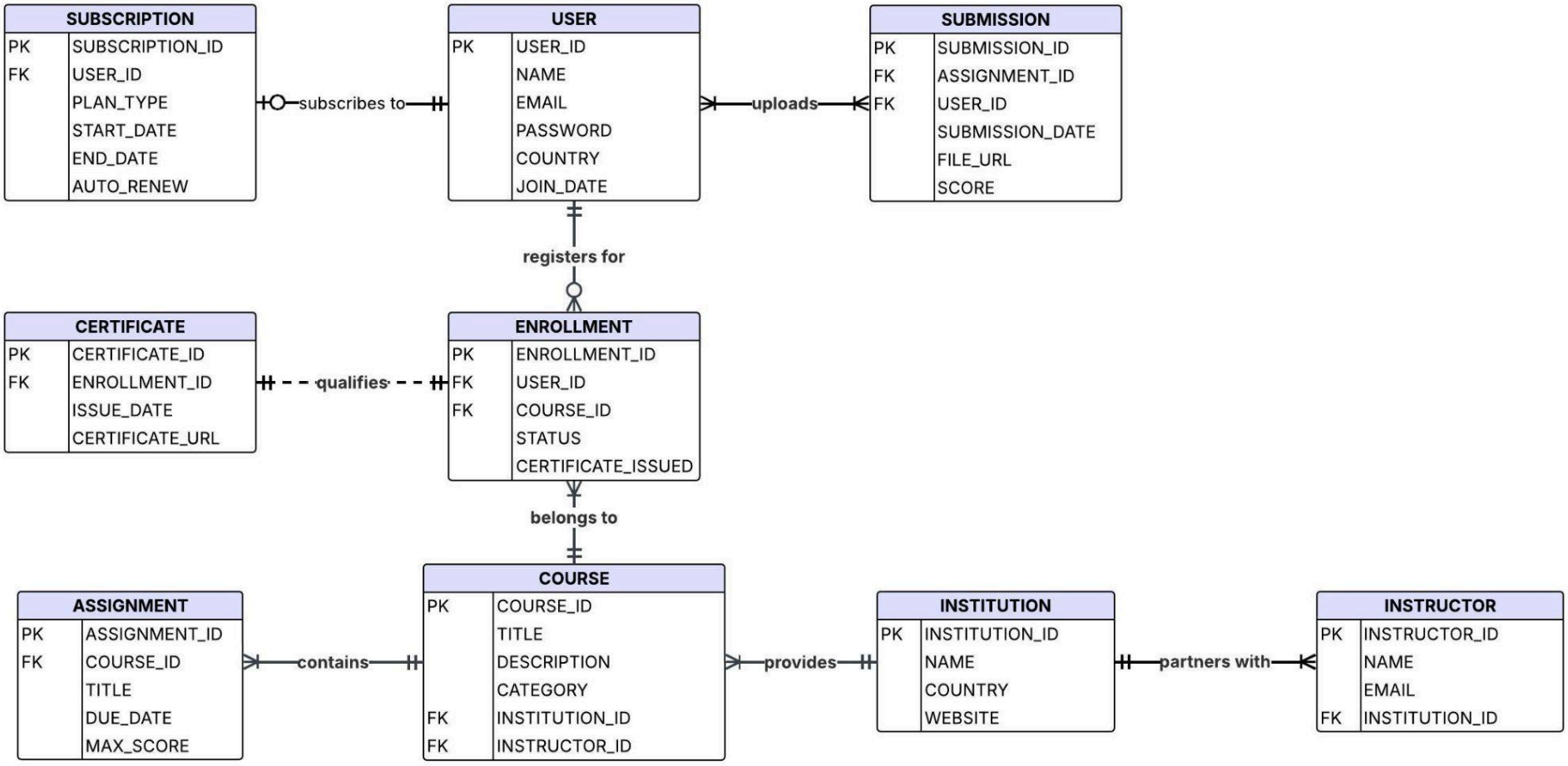
Entity	Attributes
USER	USER_ID (PK), NAME, EMAIL, PASSWORD, COUNTRY, JOIN_DATE
COURSE	COURSE_ID (PK), TITLE, DESCRIPTION, CATEGORY, INSTITUTION_ID , INSTRUCTOR_ID
ENROLLMENT	ENROLLMENT_ID (PK), USER_ID , COURSE_ID , STATUS, CERTIFICATE_ISSUED
ASSIGNMENT	ASSIGNMENT_ID (PK), COURSE_ID , TITLE, DUE_DATE, MAX_SCORE
SUBMISSION	SUBMISSION_ID (PK), ASSIGNMENT_ID , USER_ID , SUBMISSION_DATE, FILE_URL, SCORE

SUBSCRIPTION	SUBSCRIPTION_ID (PK), USER_ID , PLAN_TYPE, START_DATE, END_DATE, AUTO_RENEW
CERTIFICATE	CERTIFICATE_ID (PK), ENROLLMENT_ID , ISSUE_DATE, CERTIFICATE_URL
INSTRUCTOR	INSTRUCTOR_ID (PK), NAME, EMAIL, INSTITUTION_ID
INSTITUTION	INSTITUTION_ID (PK), NAME, COUNTRY, WEBSITE

3) Normalization to Third Normal Form (3NF)

All entities in the Coursera database design are already in Third Normal Form (3NF) and do not require further normalization. Each entity satisfies First Normal Form (1NF) as all attributes are atomic, with no repeating groups or multi-valued fields. They meet Second Normal Form (2NF) because each has a single-attribute primary key, ensuring that all non-key attributes are fully functionally dependent on the primary key without any partial dependencies. Finally, Third Normal Form (3NF) is satisfied as there are no transitive dependencies - no non-key attribute depends on another non-key attribute within the same table. Foreign keys such as InstitutionID, CourseID, and UserID are used strictly to establish relationships and do not cause redundancy or violation of normalization principles. Therefore, all entities (User, Course, Instructor, Institution, Enrollment, Assignment, Submission, Certificate, and optional Subscription) are well-structured, efficient, and compliant with 3NF.

4) Entity-Relationship Diagram (ERD) Analysis – Coursera System



This ERD models a course enrollment and learning management system for an online education platform. It includes nine key entities: USER, COURSE, INSTRUCTOR, INSTITUTION, ENROLLMENT, CERTIFICATE, ASSIGNMENT, SUBMISSION, and SUBSCRIPTION. Each entity is supported by relevant attributes and relationships with proper cardinalities and constraints.

❖ ENTITY DESCRIPTIONS

1. USER

- Primary Key (PK): USER_ID
- Attributes: NAME, EMAIL, PASSWORD, COUNTRY, JOIN_DATE

2. COURSE

- Primary Key (PK): COURSE_ID
- Foreign Key (FK): INSTRUCTOR_ID, INSTITUTION_ID
- Attributes: TITLE, DESCRIPTION, CATEGORY

3. INSTRUCTOR

- Primary Key (PK): INSTRUCTOR_ID
- Foreign Key (FK): INSTITUTION_ID
- Attributes: NAME, EMAIL, INSTITUTION_ID

4. INSTITUTION

- Primary Key (PK): INSTITUTION_ID
- Attributes: NAME, COUNTRY, WEBSITE

5. ENROLLMENT (Associate Entity)

- Primary Key (PK): ENROLLMENT_ID
- Foreign Keys (FK): USER_ID, COURSE_ID
- Attributes: STATUS, CERTIFICATE_ISSUED

6. CERTIFICATE

- Primary Key (PK): CERTIFICATE_ID
- Foreign Key (FK): ENROLLMENT_ID
- Attributes: ISSUE_DATE, CERTIFICATE_URL

7. ASSIGNMENT

- Primary Key (PK): `ASSIGNMENT_ID`
- Foreign Key (FK) : `COURSE_ID`
- Attributes: `TITLE`, `DUE_DATE`, `MAX_SCORE`

8. SUBMISSION

- Primary Key (PK): `SUBMISSION_ID`
- Foreign Keys (FK): `ASSIGNMENT_ID`, `USER_ID`
- Attributes: `SUBMISSION_DATE`, `FILE_URL`, `SCORE`

9. SUBSCRIPTION

- Primary Key (PK): `SUBSCRIPTION_ID`
- Foreign Key (FK): `USER_ID`
- Attributes: `PLAN_TYPE`, `START_DATE`, `END_DATE`, `AUTO_RENEW`

❖ **RELATIONSHIP EXPLANATIONS**

- USER – ENROLLMENT

- Relationship: “registers for”
- Type: One-to-Many
- Cardinality: USER (1,1) → registers for → ENROLLMENT (0,N)
- Justification: Each user can enroll in multiple courses. Each enrollment is tied to one user. This is a strong relationship as Subscription has its own ID (Subscription_ID) independent of User_ID.

- ENROLLMENT – COURSE

- Relationship: “belongs to”
- Type: Many-to-One
- Cardinality: ENROLLMENT (0,N) → belongs to → COURSE (1,1)
- Justification: A course can have many enrolled users, each via a separate enrollment. This is a strong relationship as Enrollment has its own Enrollment_ID PK.

- ENROLLMENT – CERTIFICATE

- Relationship: “qualifies”
- Type: One-to-One (Optional)
- Cardinality: ENROLLMENT (1,1) → qualifies → CERTIFICATE (0,1)
- Justification: A certificate is issued for an enrollment only if completed.This is a weak

relationship as Certificate depends heavily on Enrollment (qualifies relationship) — no Enrollment, no Certificate.

- COURSE – ASSIGNMENT

- Relationship: “contains”
- Type: One-to-Many
- Cardinality: COURSE (1,1) → contains → ASSIGNMENT (0,N)
- Justification: Each course includes multiple assignments. This is a strong relationship as Assignment has its own Assignment_ID PK.

- USER – SUBMISSION

- Relationship: “uploads”
- Type: One-to-Many
- Cardinality: USER (1,1) → uploads → SUBMISSION (0,N)
- Justification: A user can submit many assignments.This is a strong relationship as Submission has its own Submission_ID PK.

- ASSIGNMENT – SUBMISSION

- Relationship: “receives”
- Type: One-to-Many
- Cardinality: ASSIGNMENT (1,1) → receives → SUBMISSION (0,N)
- Justification: Each assignment can have many submissions. This is a strong relationship as Submission has its own Submission_ID PK.

- USER – SUBSCRIPTION

- Relationship: “subscribes to”
- Type: One-to-One
- Cardinality: USER (1,1) → subscribes to → SUBSCRIPTION (0,1)
- Justification: A user can optionally subscribe to a plan. This is a strong relationship as Subscription has its own ID (Subscription_ID) independent of User_ID.

- INSTITUTION – COURSE

- Relationship: “provides”
- Type: One-to-Many
- Cardinality: INSTITUTION (1,1) → provides → COURSE (0,N)
- Justification: Each institution offers multiple courses. This is a strong relationship as Course has its own Course_ID PK.

- INSTITUTION – INSTRUCTOR

- Relationship: “partners with”

- Type: One-to-Many
- Cardinality: INSTITUTION (1,1) → partners with → INSTRUCTOR (0,N)
- Justification: Each instructor is affiliated with one institution. This is a strong relationship as Instructor has its own Instructor_ID PK.

- COURSE – INSTRUCTOR

- Relationship: “teaches”
- Type: Many-to-One
- Cardinality: COURSE (0,N) → teaches → INSTRUCTOR (1,1)
- Justification: Each course is taught by one instructor, but an instructor can teach many courses.This is a strong relationship as Course has its own Course_ID PK.

10 Basic SQL

Learn basic SQL commands

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

Outcome	Weight
Reflection	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

Outcome	Weight
Research and critical review	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

Date	Author	Comment
2025/04/04 11:39	Thi Ngo	Working On It
2025/04/06 23:12	Thi Ngo	Ready to Mark
2025/04/13 01:40	Mohammad Hossain	Well-done.
2025/04/13 01:40	Mohammad Hossain	Complete
2025/04/13 14:14	Thi Ngo	Many thanks Dr. Mohammad for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Basic SQL

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/06 23:12

Tutor:
Mohammad belayet Hossain

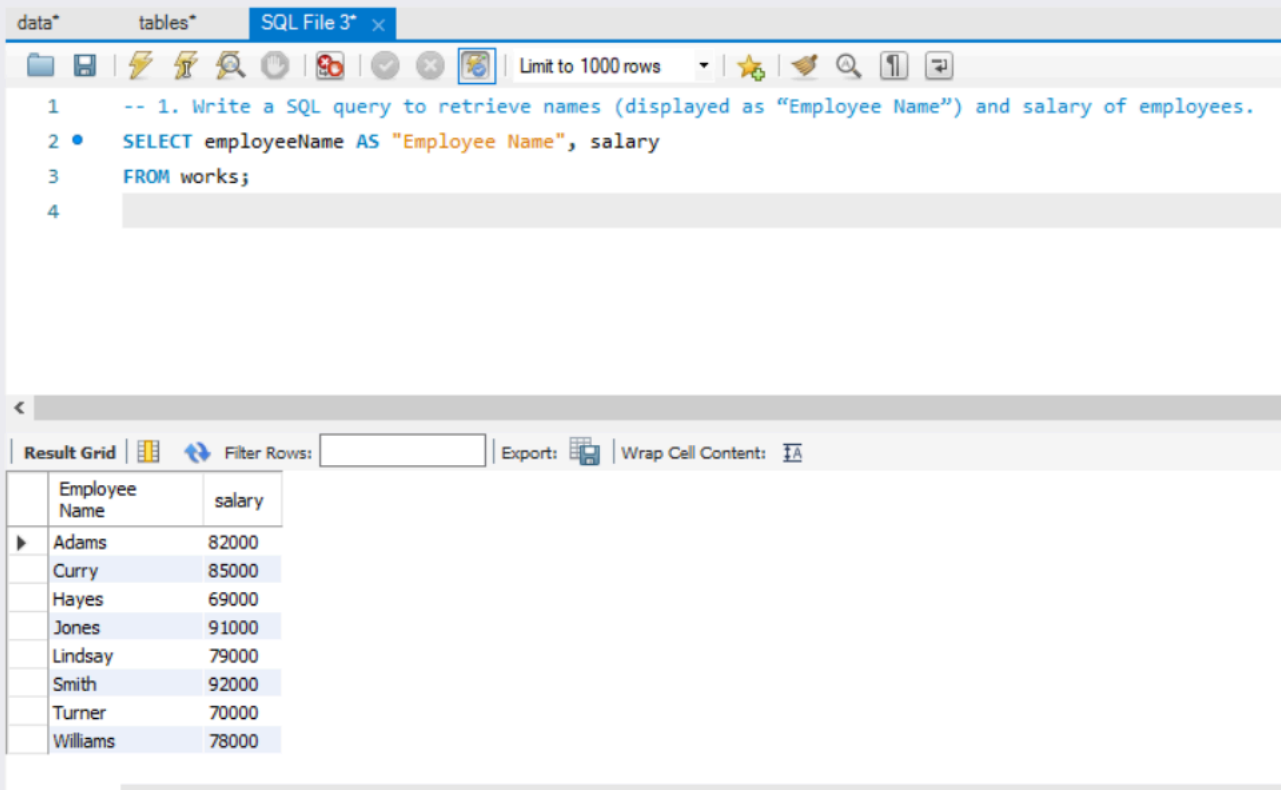
Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

In this task, I learned how to write basic SQL queries using MySQL Workbench to retrieve and analyze data from single tables. I practiced using the SELECT statement with various clauses such as WHERE, ORDER BY, DISTINCT, and GROUP BY, and worked with conditions, sorting, and subqueries. I also gained experience creating and inserting data into tables, and using aliases to simplify complex queries. Overall, this task helped me build a strong foundation in querying relational databases and understanding how to extract meaningful information from structured data.

April 6, 2025

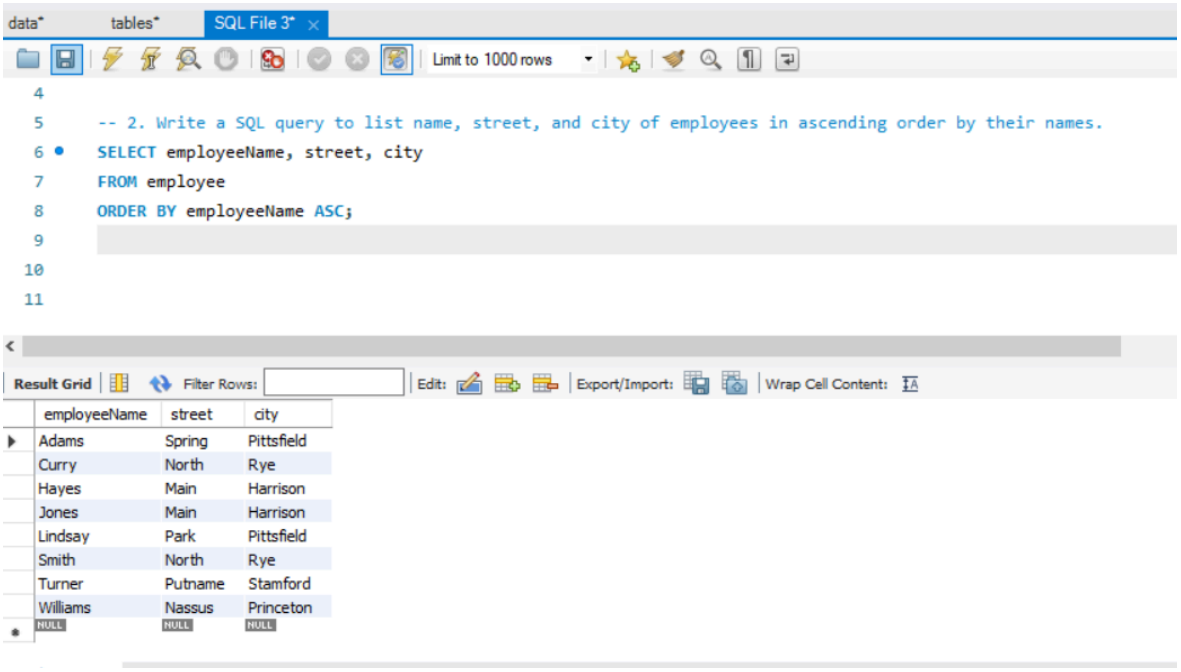
1. Write a SQL query to retrieve names (displayed as “Employee Name”) and salary of employees.

```
SELECT employeeName AS "Employee Name", salary
FROM works;
```



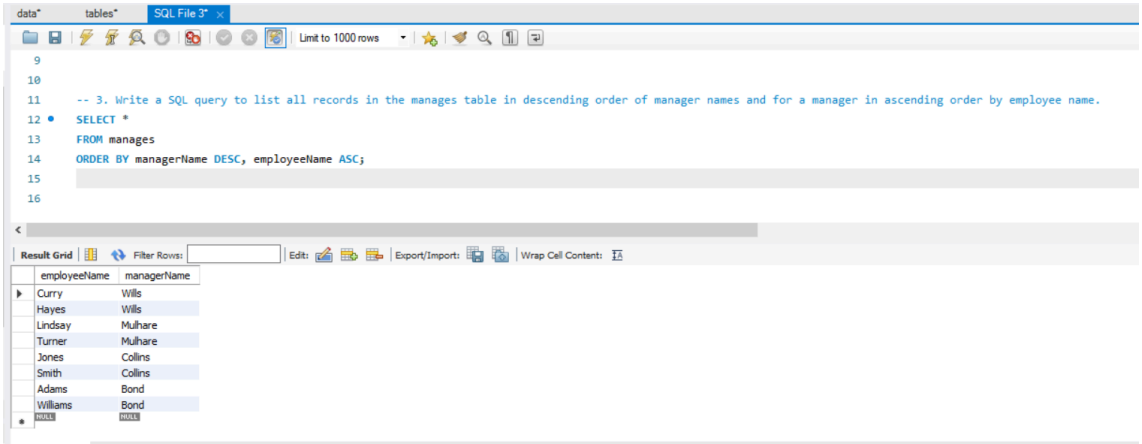
2. Write a SQL query to list name, street, and city of employees in ascending order by their names.

```
SELECT employeeName, street, city
FROM employee
ORDER BY employeeName ASC;
```



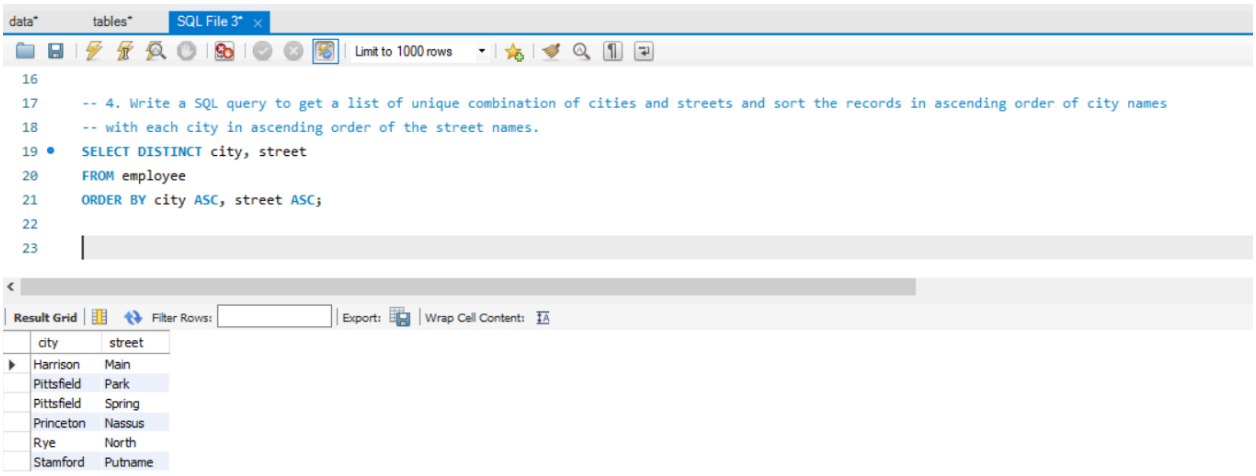
3. Write a SQL query to list all records in the manages table in descending order of manager names and for a manager in ascending order by employee name.

SELECT *
FROM manages
ORDER BY managerName DESC, employeeName ASC;



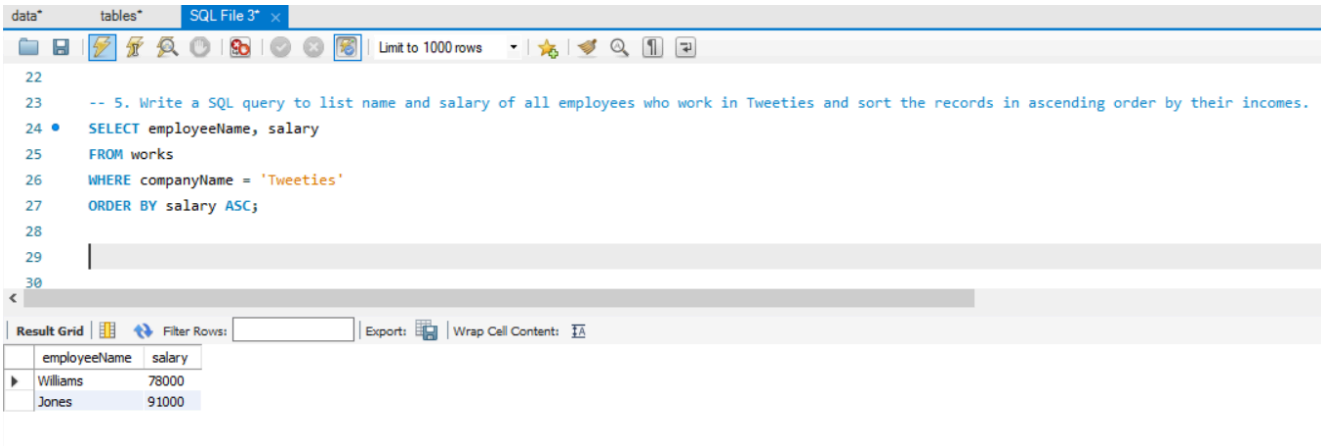
4. Write a SQL query to get a list of unique combination of cities and streets and sort the records in ascending order of city names with each city in ascending order of the street names.

```
SELECT DISTINCT city, street
FROM employee
ORDER BY city ASC, street ASC;
```



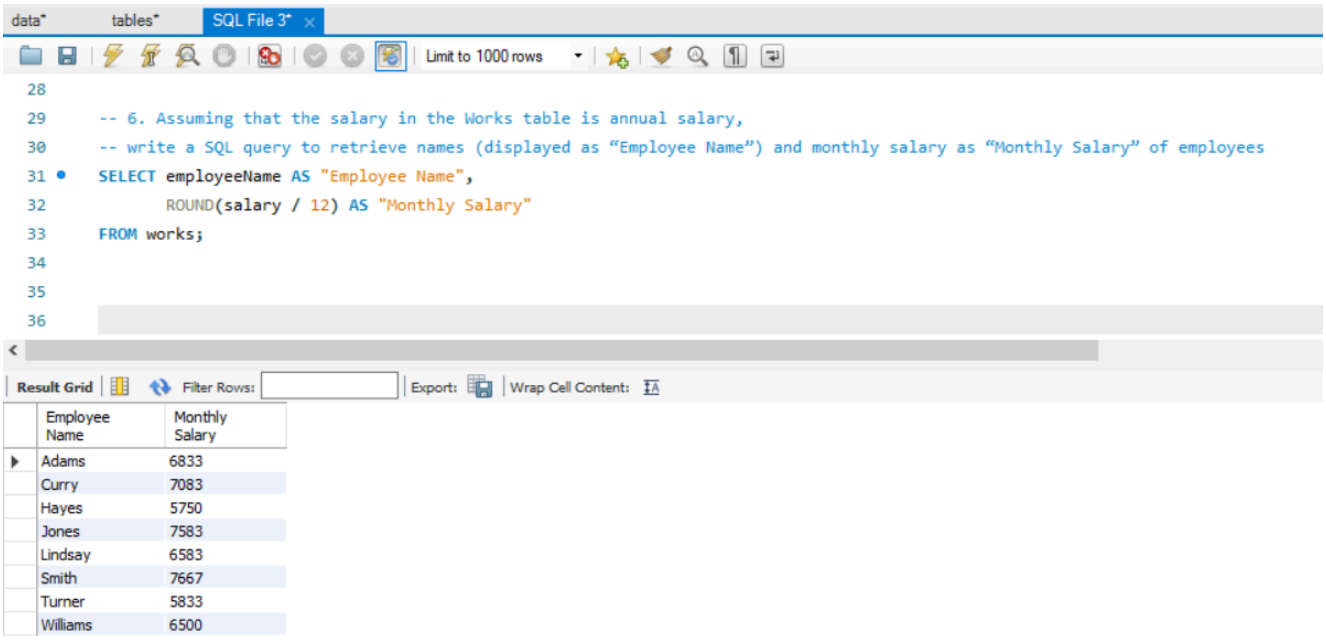
5. Write a SQL query to list name and salary of all employees who work in Tweeties and sort the records in ascending order by their incomes.

```
SELECT employeeName, salary
FROM works
WHERE companyName = 'Tweeties'
ORDER BY salary ASC;
```



6. Assuming that the salary in the Works table is annual salary, write a SQL query to retrieve names (displayed as “Employee Name”) and monthly salary as “Monthly Salary” of employees.

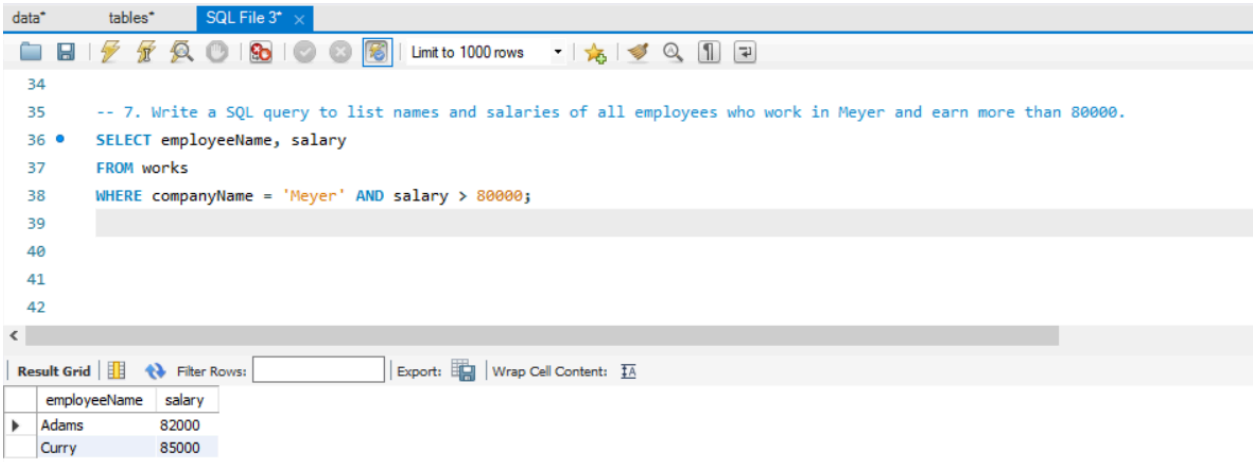
```
SELECT employeeName AS "Employee Name",
      ROUND(salary / 12) AS "Monthly Salary"
FROM works;
```



Employee Name	Monthly Salary
Adams	6833
Curry	7083
Hayes	5750
Jones	7583
Lindsay	6583
Smith	7667
Turner	5833
Williams	6500

7. Write a SQL query to list names and salaries of all employees who work in Meyer and earn more than 80000.

```
SELECT employeeName, salary
FROM works
WHERE companyName = 'Meyer' AND salary > 80000;
```



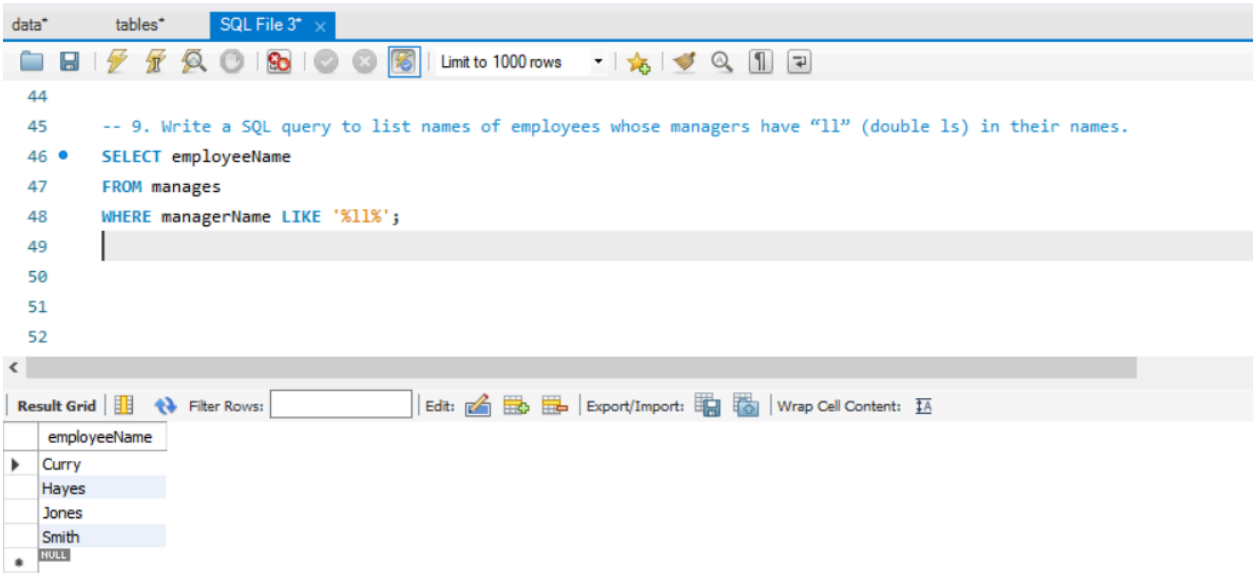
employeeName	salary
Adams	82000
Curry	85000

8. Write a SQL query to list names and companies of the employees who earn in the range of 80000 to 85000.

```
SELECT employeeName, companyName
FROM works
WHERE salary BETWEEN 80000 AND 85000;
```

9. Write a SQL query to list names of employees whose managers have “ll” (double ls) in their names.

```
SELECT employeeName
FROM manages
WHERE managerName LIKE '%ll%';
```



10. Write a SQL query to get details of employees who earns the highest in each company.

```
SELECT employeeName, companyName, salary
```

```
FROM works w
WHERE salary = (
   SELECT MAX(salary)
   FROM works
   WHERE companyName = w.companyName
);
```

```
51 •  SELECT employeeName, companyName, salary
52    FROM works w
53    WHERE salary = (
54        SELECT MAX(salary)
55        FROM works
56        WHERE companyName = w.companyName
57    );
58
59
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employeeName	companyName	salary
Curry	Meyer	85000
Hayes	Woolworths	69000
Jones	Tweeties	91000
Smith	Waltons	92000
Turner	Firebrand	70000
NULL	NULL	NULL

11. Write a SQL query to list the name of the companies with average salary of employees more than or equal to 84000.

```
SELECT companyName
FROM works
GROUP BY companyName
HAVING AVG(salary) >= 84000;
```

data* tables* SQL File 3*

Limit to 1000 rows

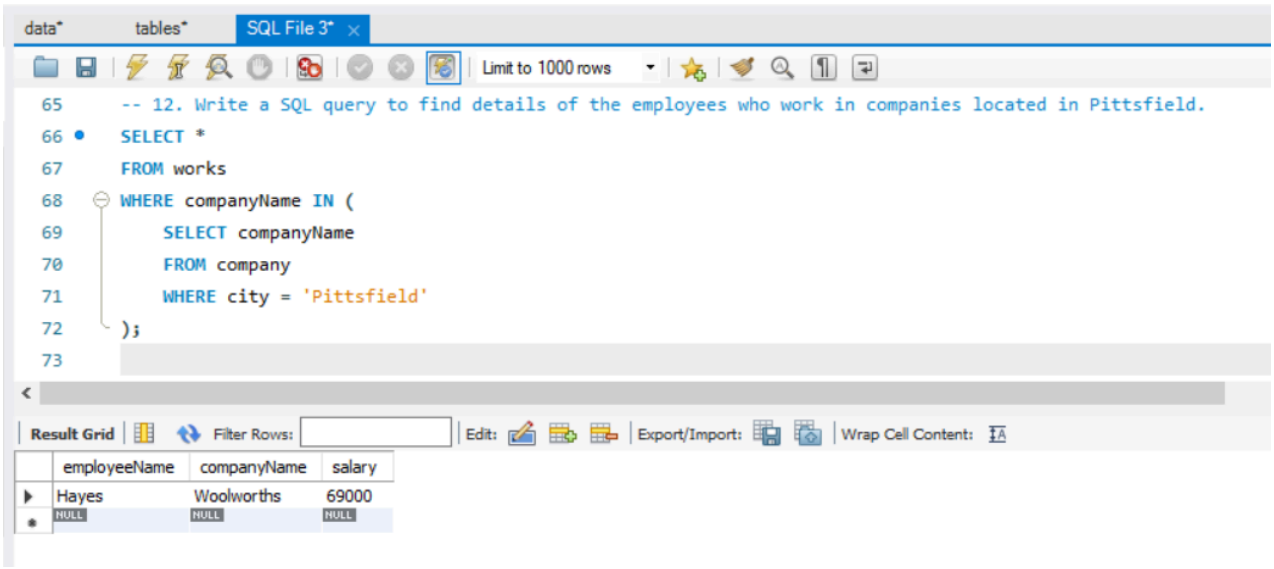
```
58
59    -- 11. Write a SQL query to list the name of the companies with average salary of employees more than or equal to 84000.
60 •  SELECT companyName
61    FROM works
62    GROUP BY companyName
63    HAVING AVG(salary) >= 84000;
64
65
66
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

companyName
Tweeties
Waltons

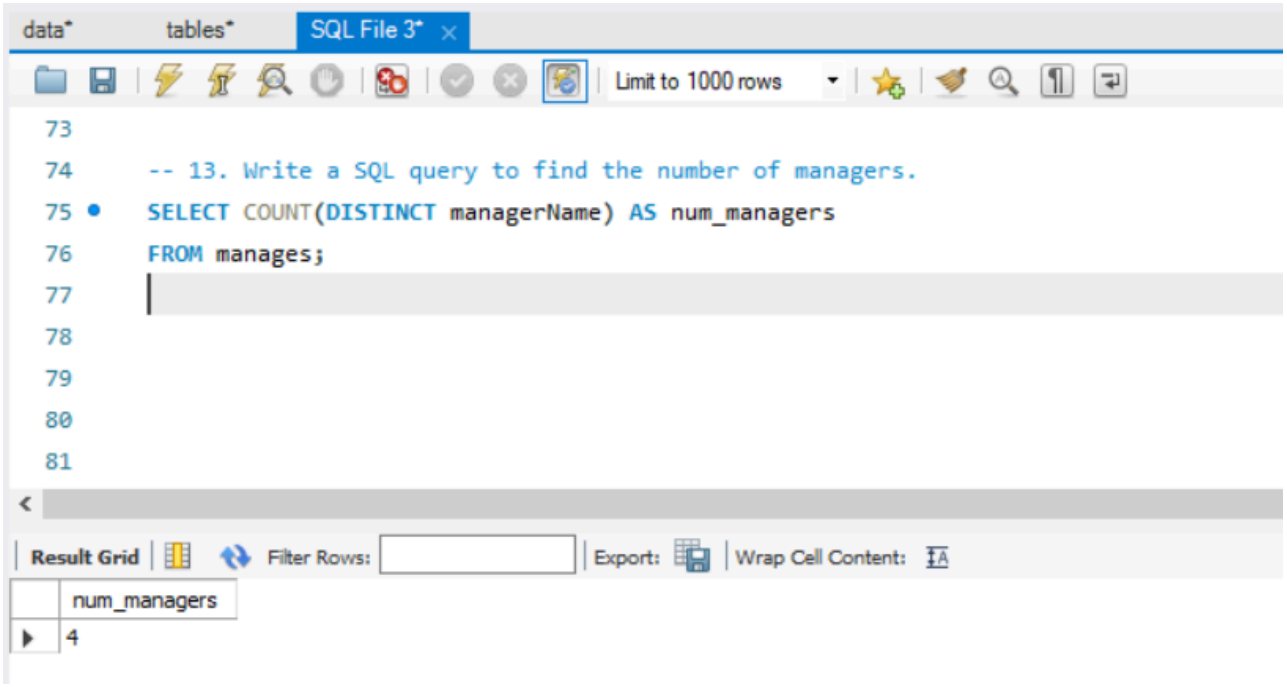
12. Write a SQL query to find details of the employees who work in companies located in Pittsfield.

```
SELECT *
FROM works
WHERE companyName IN (
   SELECT companyName
   FROM company
   WHERE city = 'Pittsfield'
);
```



13. Write a SQL query to find the number of managers.

```
SELECT COUNT(DISTINCT managerName) AS num_managers
FROM manages;
```


data*
tables*
SQL File 3*
Limit to 1000 rows
73
74 -- 13. Write a SQL query to find the number of managers.
75 SELECT COUNT(DISTINCT managerName) AS num_managers
76 FROM manages;
77
78
79
80
81
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
num_managers
4

11 Online Quiz 1

Complete Online Quiz 1 available on the unit site

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

Outcome	Weight
Reflection	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

Outcome	Weight
Research and critical review	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

Date	Author	Comment
2025/04/04 11:39	Thi Ngo	Working On It
2025/04/06 18:06	Thi Ngo	Ready to Mark
2025/04/13 14:34	Mohammad Hossain	Excellent work.
2025/04/13 14:35	Mohammad Hossain	Complete
2025/04/14 16:37	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Online Quiz 1

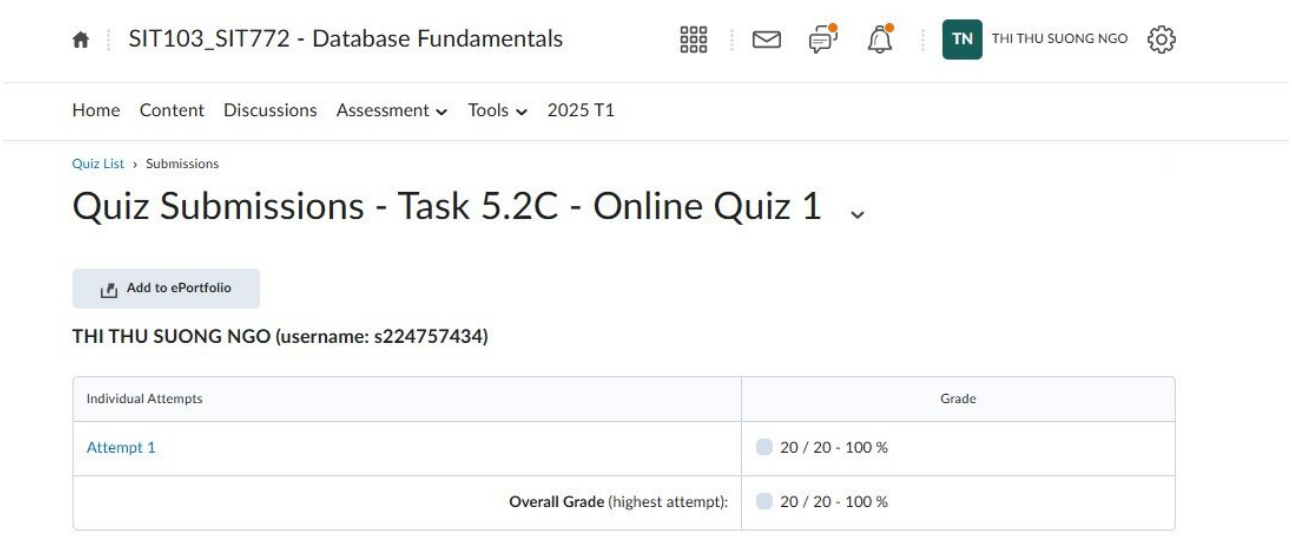
Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/06 18:06

Tutor:
Mohammad belayet Hossain

Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

From this task, I learned the fundamental concepts of databases, including key terms like data, information, metadata, and normalization. I gained an understanding of how data is structured in relational databases, how relationships are formed between tables using primary and foreign keys, and the importance of concepts such as data redundancy, integrity, and constraints. I also learned about the normalization process, its purpose in reducing redundancy, and that for most practical business applications, reaching Third Normal Form (3NF) is sufficient. Overall, this task helped reinforce my knowledge of database design and terminology.

April 6, 2025


SIT103_SIT772 - Database Fundamentals
TN
THI THU SUONG NGO
Home
Content
Discussions
Assessment
Tools
2025 T1
Quiz List › Submissions
Quiz Submissions - Task 5.2C - Online Quiz 1
Add to ePortfolio
THI THU SUONG NGO (username: s224757434)
Individual Attempts
Grade
Attempt 1
20 / 20 - 100 %
Overall Grade (highest attempt):
20 / 20 - 100 %

12 SELECT Queries with JOIN

Write different types of SELECT queries to extract required information from a database

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

Outcome	Weight
Reflection	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

Outcome	Weight
Research and critical review	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

Date	Author	Comment
2025/04/11 17:11	Thi Ngo	Working On It
2025/04/23 16:12	Thi Ngo	Ready to Mark
2025/04/23 16:18	Thi Ngo	Working On It
2025/04/23 16:18	Thi Ngo	Working On It
2025/04/23 16:18	Thi Ngo	Working On It
2025/04/23 16:19	Thi Ngo	Ready to Mark
2025/04/25 14:55	Mohammad Hossain	Good work. However, for Q3 the attribute patron_ID is missing.
2025/04/25 14:55	Mohammad Hossain	Complete
2025/05/19 17:46	Thi Ngo	Thank you very much for your feedback, I have noted on it.

Deakin University

Database Fundamentals

OnTrack Submission

SELECT Queries with JOIN

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/23 16:19

Tutor:
Mohammad belayet Hossain

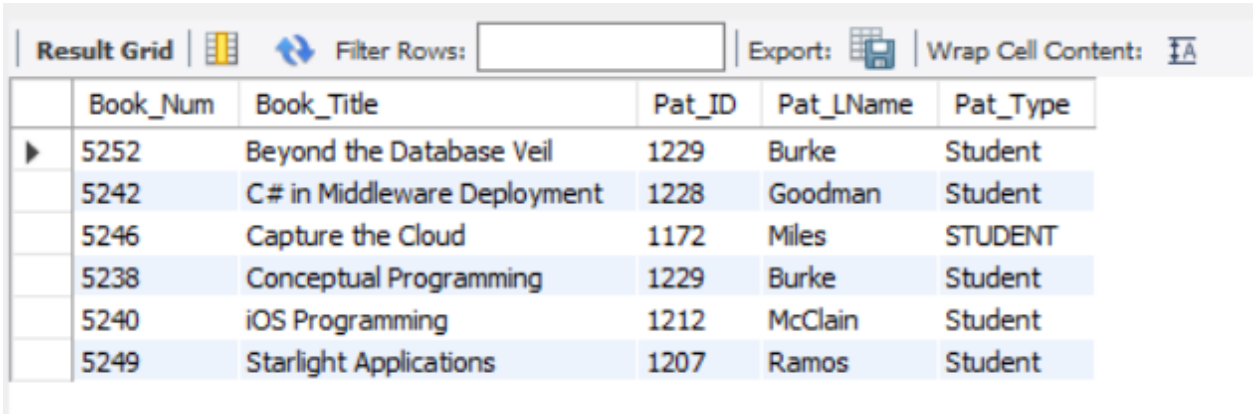
Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

From this task, I learned how to write advanced SQL queries using JOIN, GROUP BY, HAVING, and SQL functions like AVG(), COUNT(), and CONCAT(). I gained a deeper understanding of how to retrieve and filter data across multiple related tables based on specific business rules. In particular, I learned how to differentiate between filtering raw rows (WHERE) and grouped results (HAVING), how to handle partial data (like missing Check_In_Date), and how to use conditional logic in aggregation with CASE WHEN. This task also helped reinforce the importance of accurately interpreting requirements and aligning output formats to expected results.

April 23, 2025

1) **Write a SQL statement to display the book number, title, patron ID, last name, and patron type for books currently checked out for a patron. Sort the result by book title.**

```
SELECT
   B.Book_Num,
   B.Book_Title,
   P.Pat_ID,
   P.Pat_LName,
   P.Pat_Type
FROM
   BOOK B
JOIN
   CHECKOUT C ON B.Book_Num = C.Book_Num
JOIN
   PATRON P ON C.Pat_ID = P.Pat_ID
WHERE
   C.Check_In_Date IS NULL
ORDER BY
   B.Book_Title;
```

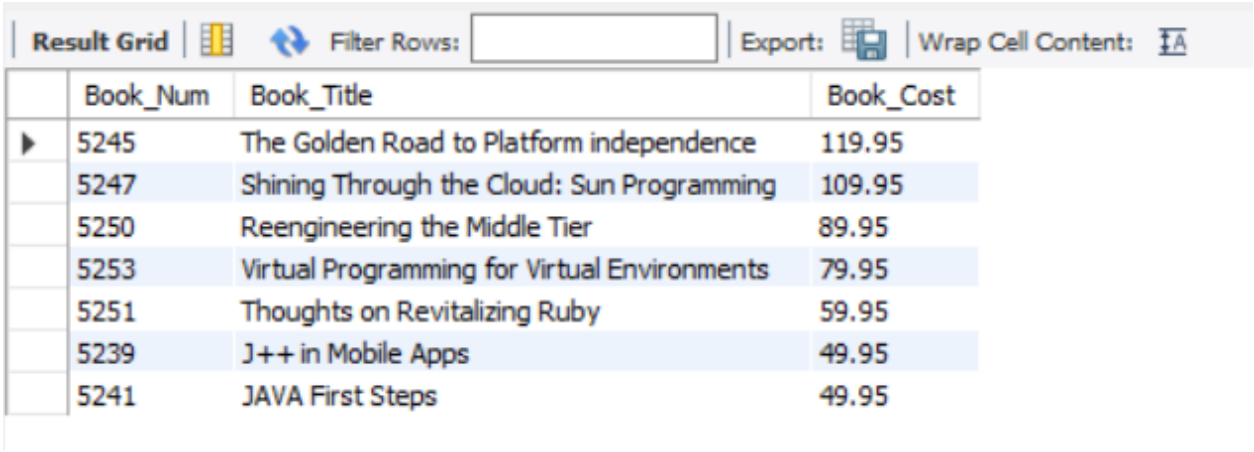


Book_Num	Book_Title	Pat_ID	Pat_LName	Pat_Type
5252	Beyond the Database Veil	1229	Burke	Student
5242	C# in Middleware Deployment	1228	Goodman	Student
5246	Capture the Cloud	1172	Miles	STUDENT
5238	Conceptual Programming	1229	Burke	Student
5240	iOS Programming	1212	McClain	Student
5249	Starlight Applications	1207	Ramos	Student

2) **Write a SQL statement to display book number, title, and cost of all books that have never been checked out. Sort the results by cost in descending order and then by book title in ascending order.**

```
SELECT
   B.Book_Num,
   B.Book_Title,
   B.Book_Cost
FROM
   BOOK B
LEFT JOIN
```

```
    CHECKOUT C ON B.Book_Num = C.Book_Num
WHERE
    C.Book_Num IS NULL
ORDER BY
    B.Book_Cost DESC,
    B.Book_Title ASC;
```

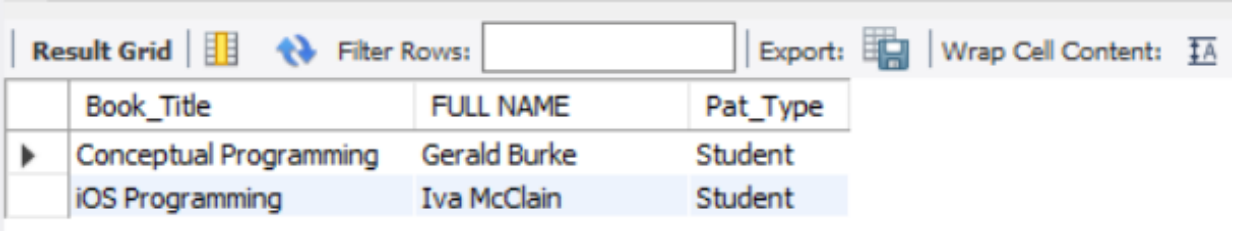


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_Num	Book_Title	Book_Cost
5245	The Golden Road to Platform independence	119.95
5247	Shining Through the Cloud: Sun Programming	109.95
5250	Reengineering the Middle Tier	89.95
5253	Virtual Programming for Virtual Environments	79.95
5251	Thoughts on Revitalizing Ruby	59.95
5239	J++ in Mobile Apps	49.95
5241	JAVA First Steps	49.95

3) Write a SQL statement to display the book title, patron ID, full name (first and last), and patron type for patrons who currently have a book checked out in the subject "Programming." Sort the result by book title and then by last name and first name.

```
SELECT
    B.Book_Title,
    CONCAT(P.Pat_FName, ' ', P.Pat_LName) AS `FULL NAME`,
    P.Pat_Type
FROM
    BOOK B
JOIN
    CHECKOUT C ON B.Book_Num = C.Book_Num
JOIN
    PATRON P ON C.Pat_ID = P.Pat_ID
WHERE
    C.Check_In_Date IS NULL
    AND B.Book_Subject = 'Programming'
ORDER BY
    B.Book_Title,
    P.Pat_LName,
    P.Pat_FName;
```

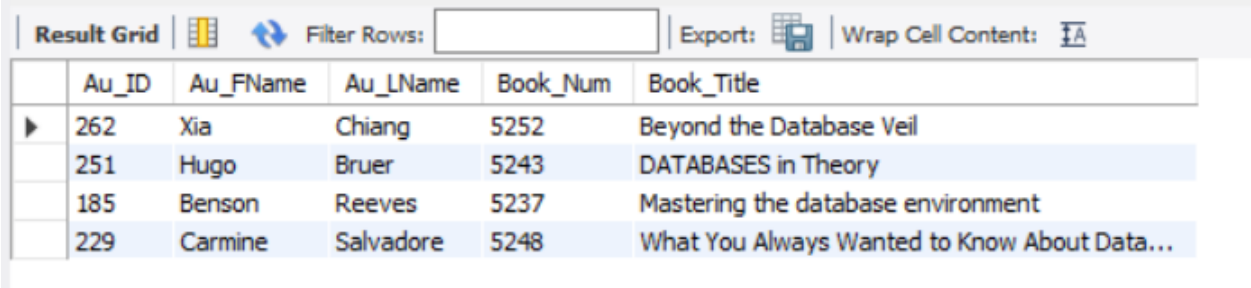


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_Title	FULL NAME	Pat_Type
Conceptual Programming	Gerald Burke	Student
iOS Programming	Iva McClain	Student

4) Write a SQL statement to list author ID, first and last name, book number and title of all books in the subject "Database." Sort the results by book title in ascending order and then by author last name in ascending order.

```
SELECT
   A.Au_ID,
   A.Au_FName,
   A.Au_LName,
   B.Book_Num,
   B.Book_Title
FROM
   AUTHOR A
JOIN
   WRITES W ON A.Au_ID = W.Au_ID
JOIN
   BOOK B ON W.Book_Num = B.Book_Num
WHERE
   B.Book_Subject = 'Database'
ORDER BY
   B.Book_Title ASC,
   A.Au_LName ASC;
```



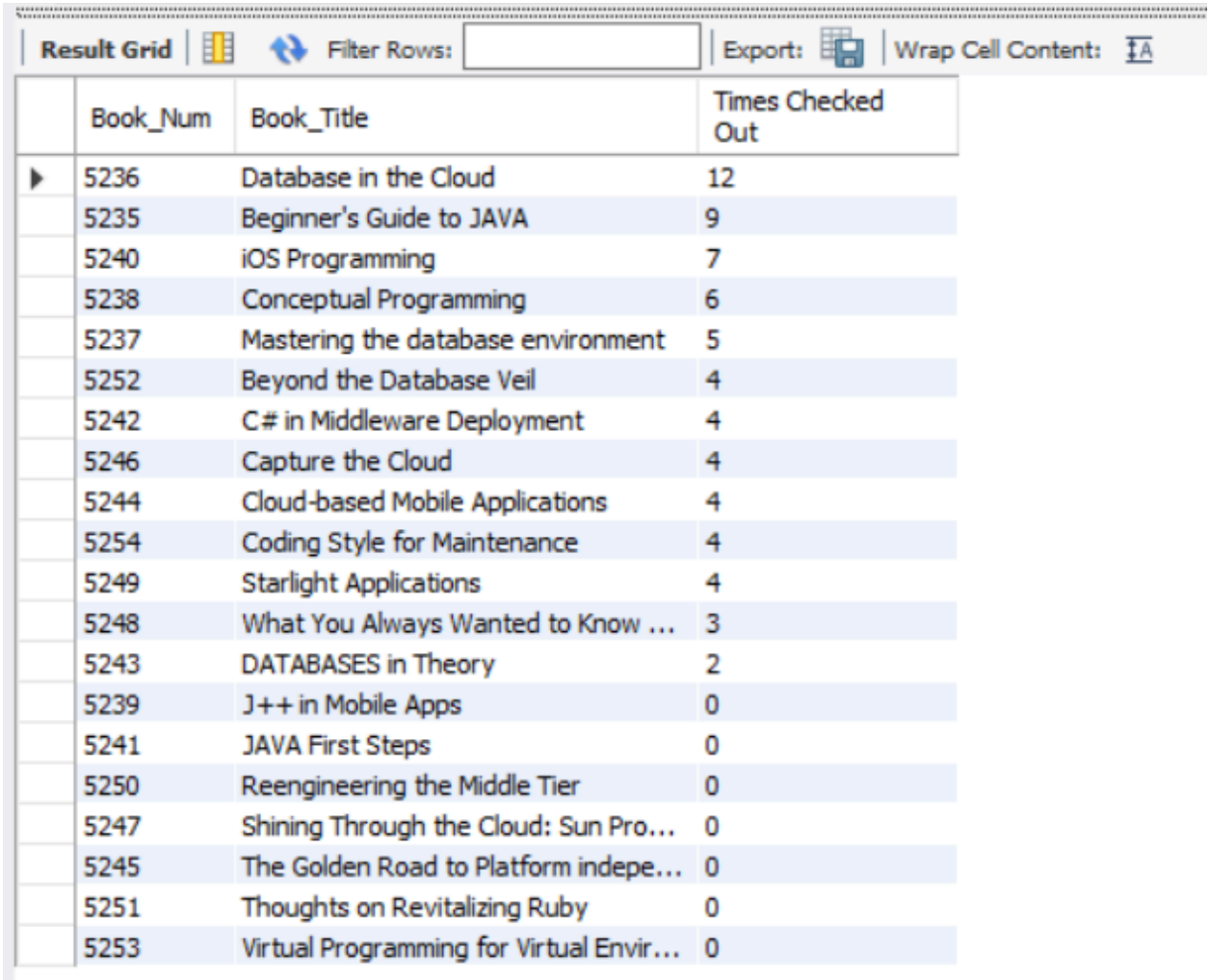
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Au_ID	Au_FName	Au_LName	Book_Num	Book_Title
262	Xia	Chiang	5252	Beyond the Database Veil
251	Hugo	Bruer	5243	DATABASES in Theory
185	Benson	Reeves	5237	Mastering the database environment
229	Carmine	Salvadore	5248	What You Always Wanted to Know About Data…

5) Write a SQL statement to display book number, title, and number of times each book has been checked out (column name as 'Times Checked Out'). Sort the results in descending order by the number of times checked out and then by title in ascending order.

```
SELECT
   B.Book_Num,
```

```
    B.Book_Title,
    COUNT(C.Check_Num) AS `Times Checked Out`
FROM
    BOOK B
LEFT JOIN
    CHECKOUT C ON B.Book_Num = C.Book_Num
GROUP BY
    B.Book_Num, B.Book_Title
ORDER BY
    `Times Checked Out` DESC,
    B.Book_Title ASC;
```

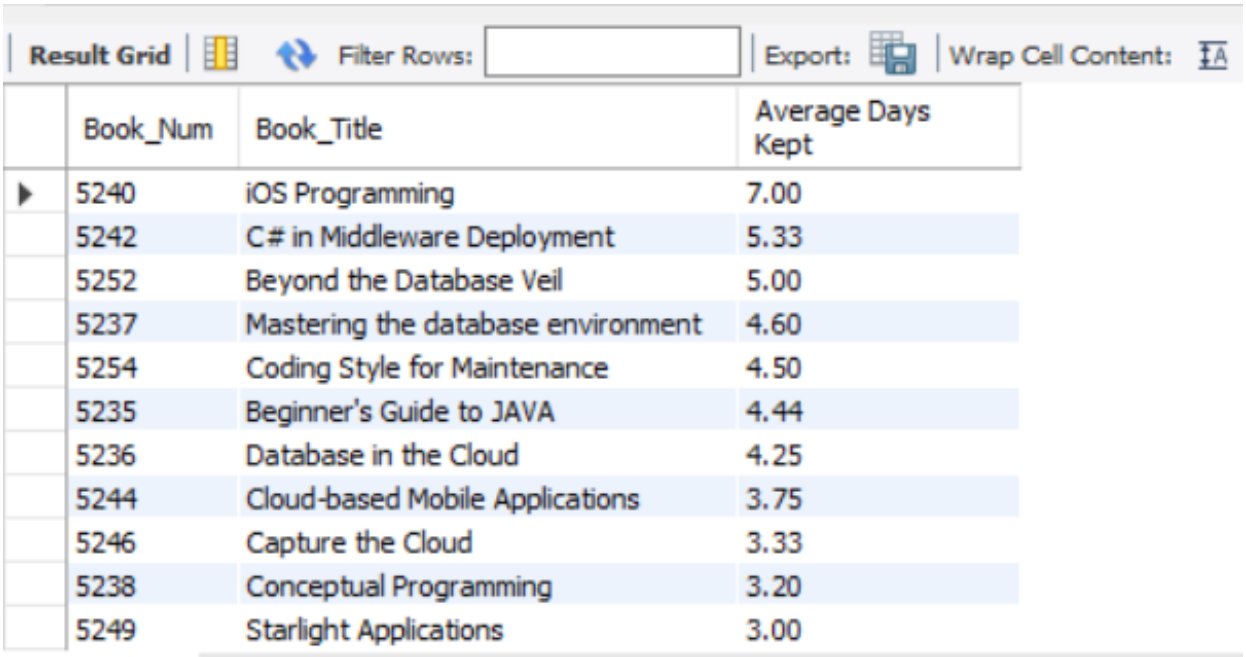


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_Num	Book_Title	Times Checked Out
5236	Database in the Cloud	12
5235	Beginner's Guide to JAVA	9
5240	iOS Programming	7
5238	Conceptual Programming	6
5237	Mastering the database environment	5
5252	Beyond the Database Veil	4
5242	C# in Middleware Deployment	4
5246	Capture the Cloud	4
5244	Cloud-based Mobile Applications	4
5254	Coding Style for Maintenance	4
5249	Starlight Applications	4
5248	What You Always Wanted to Know ...	3
5243	DATABASES in Theory	2
5239	J++ in Mobile Apps	0
5241	JAVA First Steps	0
5250	Reengineering the Middle Tier	0
5247	Shining Through the Cloud: Sun Pro...	0
5245	The Golden Road to Platform indepe...	0
5251	Thoughts on Revitalizing Ruby	0
5253	Virtual Programming for Virtual Envir...	0

6) Write a SQL statement to display book number, title, and average number of days each book is kept during a checkout (column name as 'Average Days Kept' and maintain 2 decimals numbers after calculation). Limit the results to only books which have been checked out at least 4 times. Sort the results by average days a book is kept in descending order.

```
SELECT
   B.Book_Num,
   B.Book_Title,
   ROUND(
      AVG(
         CASE
            WHEN C.Check_In_Date IS NOT NULL
            THEN DATEDIFF(C.Check_In_Date, C.Check_Out_Date)
         END
      ),
      2
   ) AS `Average Days Kept`
FROM
   BOOK B
JOIN
   CHECKOUT C ON B.Book_Num = C.Book_Num
GROUP BY
   B.Book_Num, B.Book_Title
HAVING
   COUNT(*) >= 4
ORDER BY
   `Average Days Kept` DESC;
```



Book_Num	Book_Title	Average Days Kept
5240	iOS Programming	7.00
5242	C# in Middleware Deployment	5.33
5252	Beyond the Database Veil	5.00
5237	Mastering the database environment	4.60
5254	Coding Style for Maintenance	4.50
5235	Beginner's Guide to JAVA	4.44
5236	Database in the Cloud	4.25
5244	Cloud-based Mobile Applications	3.75
5246	Capture the Cloud	3.33
5238	Conceptual Programming	3.20
5249	Starlight Applications	3.00

13 Other DML and DDL Queries

Write other data manipulation and data definition queries

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

Outcome	Weight
Reflection	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

Date	Author	Comment
2025/04/17 14:34	Thi Ngo	Working On It
2025/04/30 10:00	Thi Ngo	Ready to Mark
2025/05/04 14:24	Mohammad Hossain	Good work.
2025/05/04 14:24	Mohammad Hossain	Complete
2025/05/19 17:46	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Other DML and DDL Queries

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/04/30 10:00

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through this task, I learned how to use essential DDL and DML SQL statements, including creating tables with constraints, altering table structures, adding self-referencing foreign keys, inserting multiple records, updating specific data fields, and deleting records based on conditions. I also gained hands-on experience with foreign key constraints, which reinforced the importance of data order and integrity when dealing with relational databases.

April 30, 2025

Question 1) SQL statement create the EMPLOYEE table with the following attributes and constraints

(i) SQL statement to create the EMPLOYEE table

```
CREATE TABLE EMPLOYEE (
   EMP_NUM INT PRIMARY KEY,
   EMP_TITLE VARCHAR(5),
   EMP_LNAME VARCHAR(25) NOT NULL,
   EMP_FNAME VARCHAR(25) NOT NULL,
   EMP_INITIAL CHAR(1) NOT NULL,
   EMP_DOB DATE,
   EMP_HIRE_DATE DATE,
   EMP_AREACODE CHAR(3) DEFAULT '001',
   EMP_PHONE CHAR(8),
   EMP_MGR INT
);
```

(ii) A screenshot of the result of the following SQL statement: DESC EMPLOYEE;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

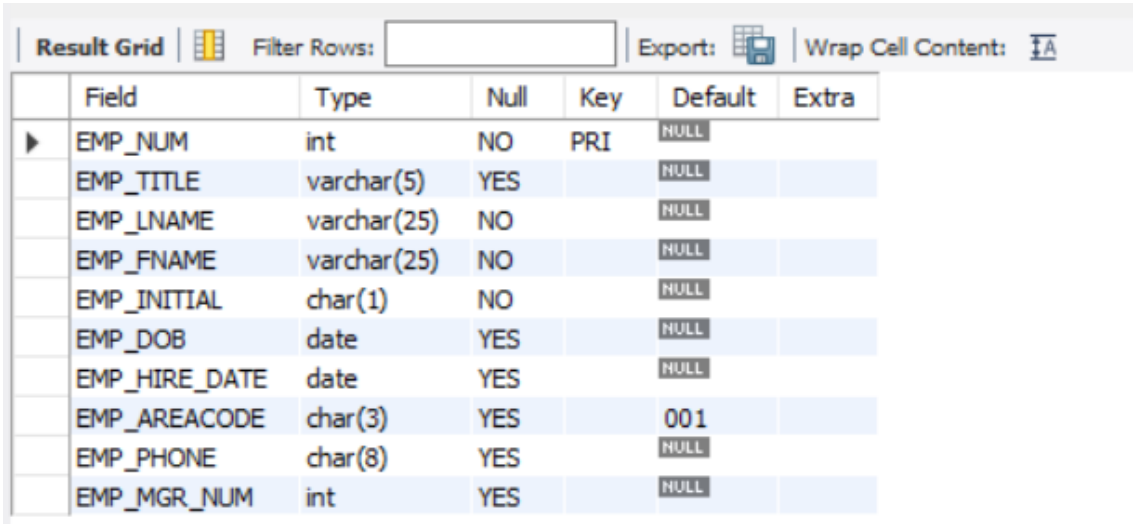
Field	Type	Null	Key	Default	Extra
EMP_NUM	int	NO	PRI	NULL	
EMP_TITLE	varchar(5)	YES		NULL	
EMP_LNAME	varchar(25)	NO		NULL	
EMP_FNAME	varchar(25)	NO		NULL	
EMP_INITIAL	char(1)	NO		NULL	
EMP_DOB	date	YES		NULL	
EMP_HIRE_DATE	date	YES		NULL	
EMP_AREACODE	char(3)	YES		001	
EMP_PHONE	char(8)	YES		NULL	
EMP_MGR	int	YES		NULL	

Question 2) SQL statement to change the column name 'EMP_MGR' to 'EMP_MGR_NUM

(i) SQL statement to alter the EMPLOYEE table

```
ALTER TABLE EMPLOYEE
CHANGE EMP_MGR EMP_MGR_NUM INT;
```

ii) A screenshot of the result of the following SQL statement: DESC EMPLOYEE;



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
EMP_NUM	int	NO	PRI	NULL	
EMP_TITLE	varchar(5)	YES		NULL	
EMP_LNAME	varchar(25)	NO		NULL	
EMP_FNAME	varchar(25)	NO		NULL	
EMP_INITIAL	char(1)	NO		NULL	
EMP_DOB	date	YES		NULL	
EMP_HIRE_DATE	date	YES		NULL	
EMP_AREACODE	char(3)	YES		001	
EMP_PHONE	char(8)	YES		NULL	
EMP_MGR_NUM	int	YES		NULL	

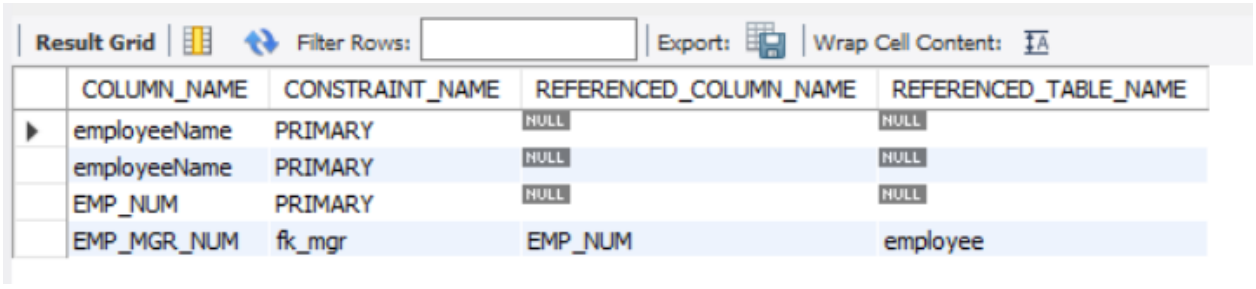
Question 3) SQL statement to ADD FOREIGN KEY CONSTRAINT to EMP_MGR_NUM to reference to EMP_NUM

(i) SQL statement to add the FOREIGN CONSTRAINT

```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT fk_mgr
FOREIGN KEY (EMP_MGR_NUM) REFERENCES EMPLOYEE(EMP_NUM);
```

(ii) A screenshot of the result of the following SQL statement

```
SELECT COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_COLUMN_NAME,
REFERENCED_TABLE_NAME
FROM information_schema.KEY_COLUMN_USAGE
WHERE TABLE_NAME = 'EMPLOYEE';
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_COLUMN_NAME	REFERENCED_TABLE_NAME
employeeName	PRIMARY	NULL	NULL
employeeName	PRIMARY	NULL	NULL
EMP_NUM	PRIMARY	NULL	NULL
EMP_MGR_NUM	fk_mgr	EMP_NUM	employee

Question 4) SQL statements to insert the following records in the EMPLOYEE TABLE.

```
INSERT INTO EMPLOYEE (
  EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL,
  EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM
```

```
) VALUES
-- Top-level managers (no manager assigned)
(100, 'Mr.', 'Kolmycz', 'George', 'D', '1912-06-15', '1985-03-15', '615', '321-5455', NULL),
(105, 'Mr.', 'Williams', 'Robert', 'N', '1975-03-14', '1996-11-08', '615', '320-3242', NULL),
(108, 'Mr.', 'Wiesenbach', 'Paul', 'R', '1966-02-14', '1992-11-18', '615', '897-3353', NULL),

-- Employees managed by 100
(101, 'Mrs.', 'Lewis', 'Rhonda', 'G', '1965-03-19', '1986-04-25', '615', '324-4472', 100),
(102, 'Mr.', 'Vondam', 'Rhett', 'N', '1958-11-14', '1950-12-20', '901', '657-8923', 100),
(103, 'Mr.', 'Joncs', 'Anne', 'C', '1974-10-05', '1994-08-20', '615', '895-4894', 100),

-- Employees managed by 105
(104, 'Mrs.', 'Lange', 'John', 'D', '1971-11-08', '1994-10-20', '901', '504-4430', 105),
(106, 'Mrs.', 'Smith', 'Jeanine', 'K', '1968-02-12', '1996-01-05', '615', '324-7833', 105),
(107, 'Mr.', 'Diante', 'Joyce', 'R', '1974-08-21', '1974-10-22', '615', '800-4557', 105),

-- Employees managed by 108
(109, 'Mr.', 'Smith', 'George', 'D', '1961-06-18', '1989-04-14', '901', '324-5570', 108),
(110, 'Mrs.', 'Genkaiz', 'Leighla', 'W', '1970-05-19', '1990-12-01', '901', '590-0903', 108),
(111, 'Mr.', 'Washington', 'Rupert', 'G', '1953-03-20', '1953-11-21', '901', '655-6322', 108),
(112, 'Mr.', 'Johnson', 'Edward', 'E', '1961-05-14', '1993-04-16', '615', '833-4093', 108),
(113, 'Ms.', 'Smythe', 'Melanie', 'Y', '1970-09-15', '1939-05-11', '901', '902-8312', 108),
(114, 'Ms.', 'Brandon', 'Marie', 'G', '1956-11-02', '1979-11-15', '901', '832-0445', 108),
(115, 'Mr.', 'Sarluca', 'Henriette', 'R', '1972-07-25', '1993-04-23', '615', '324-5555', 108),
(116, 'Mr.', 'Smith', 'George', 'A', '1965-11-08', '1988-12-10', '615', '890-2934', 108);
```

(ii) a screenshot of the result of the following SQL statement: SELECT * FROM EMPLOYEE;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR_NUM
100	Mr.	Kolmycz	George	D	1912-06-15	1985-03-15	615	321-5455	NULL
101	Mrs.	Lewis	Rhonda	G	1965-03-19	1986-04-25	615	324-4472	100
102	Mr.	Vondam	Rhett	N	1958-11-14	1950-12-20	901	657-8923	100
103	Mr.	Joncs	Anne	C	1974-10-05	1994-08-20	615	895-4894	100
104	Mrs.	Lange	John	D	1971-11-08	1994-10-20	901	504-4430	105
105	Mr.	Williams	Robert	N	1975-03-14	1996-11-08	615	320-3242	NULL
106	Mrs.	Smith	Jeanine	K	1968-02-12	1996-01-05	615	324-7833	105
107	Mr.	Diante	Joyce	R	1974-08-21	1974-10-22	615	800-4557	105
108	Mr.	Wiesenbach	Paul	R	1966-02-14	1992-11-18	615	897-3353	NULL
109	Mr.	Smith	George	D	1961-06-18	1989-04-14	901	324-5570	108
110	Mrs.	Genkaiz	Leighla	W	1970-05-19	1990-12-01	901	590-0903	108
111	Mr.	Washington	Rupert	G	1953-03-20	1953-11-21	901	655-6322	108
112	Mr.	Johnson	Edward	E	1961-05-14	1993-04-16	615	833-4093	108
113	Ms.	Smythe	Melanie	Y	1970-09-15	1939-05-11	901	902-8312	108
114	Ms.	Brandon	Marie	G	1956-11-02	1979-11-15	901	832-0445	108
115	Mr.	Sarluca	Henriette	R	1972-07-25	1993-04-23	615	324-5555	108
116	Mr.	Smith	George	A	1965-11-08	1988-12-10	615	890-2934	108
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Question 5) SQL statement to set the manager of Paul Wiesenbach (EMP_NUM = 108) as George Kolmycz (EMP_NUM = 100).

(i) SQL statement to update the manager num

```
UPDATE EMPLOYEE
SET EMP_MGR_NUM = 100
WHERE EMP_NUM = 108;
```

(ii) a screenshot of the result of the following SQL statement: SELECT * FROM EMPLOYEE WHERE EMP_NUM = 108;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

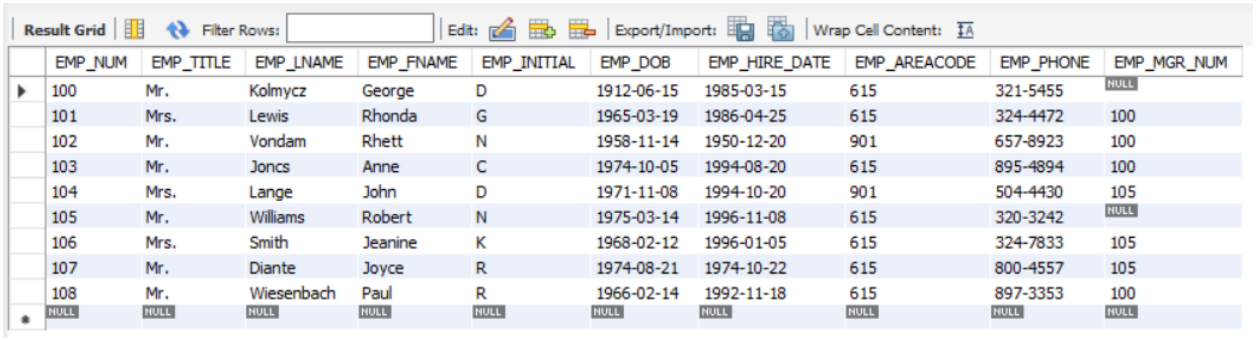
EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR_NUM
108	Mr.	Wiesenbach	Paul	R	1966-02-14	1992-11-18	615	897-3353	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Question 6) SQL statement to delete records of all employees managed by Paul Wiesenbach (EMP_NUM = 108).

(i) SQL statement to delete records

```
DELETE FROM EMPLOYEE
WHERE EMP_MGR_NUM = 108;
```

(ii) Screenshot of the result of the following SQL statement: SELECT * FROM EMPLOYEE;



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR_NUM
100	Mr.	Kolmycz	George	D	1912-06-15	1985-03-15	615	321-5455	NULL
101	Mrs.	Lewis	Rhonda	G	1965-03-19	1986-04-25	615	324-4472	100
102	Mr.	Vondam	Rhett	N	1958-11-14	1950-12-20	901	657-8923	100
103	Mr.	Joncs	Anne	C	1974-10-05	1994-08-20	615	895-4894	100
104	Mrs.	Lange	John	D	1971-11-08	1994-10-20	901	504-4430	105
105	Mr.	Williams	Robert	N	1975-03-14	1996-11-08	615	320-3242	NULL
106	Mrs.	Smith	Jeanine	K	1968-02-12	1996-01-05	615	324-7833	105
107	Mr.	Diante	Joyce	R	1974-08-21	1974-10-22	615	800-4557	105
108	Mr.	Wiesenbach	Paul	R	1966-02-14	1992-11-18	615	897-3353	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

14 Research Report and Presentation

Prepare a research report on one emerging trend or issue related to database.

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

Outcome	Weight
Reflection	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

Outcome	**Weight**
Research and critical review	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

Date	Author	Comment
2025/03/25 10:49	Thi Ngo	Working On It
2025/04/02 20:37	Thi Ngo	Not Started
2025/04/03 20:06	Thi Ngo	Working On It
2025/05/08 18:30	Thi Ngo	Ready to Mark
2025/05/08 18:46	Thi Ngo	pdf document
2025/05/08 18:46	Thi Ngo	Dear Professor,I am writing to confirm that I have submitted my presentation video for the assignment. You can access the video via the link below::link: https://youtu.be/k69zd11KN0sI've also attached the corresponding slide deck used in the presentation for your reference.Please let me know if you have any issues accessing the materials. Thank you for your support and feedback throughout this project.Best regards,Suong Ngo.
2025/05/23 13:52	Iynkaran Natgunanathan	I really appreciate your effort. However, the current submission requires major revision.While tools are mentioned (BigQuery, MLflow, Vertex AI), the report does not offer a focused case study or comparative analysis of how one DBMS has evolved with ML integration.significantly lacking contribution in terms of your own research or analysis. You can include comparison tables and informative figures generated from your own understanding and comprehension. Please clearly identify the problems, issues, or limitations in the topic and propose several suggestions to address those gaps. These suggestions should be supported by appropriate literature, including journal and conference papers, as well as credible sources such as university or reputable organization websites (e.g., Amazon).Further, the presentation should be more engaging and organized.When making changes, please use blue color to make it easier for me to review.
2025/05/23 13:52	Iynkaran Natgunanathan	Fix and Resubmit
2025/05/25 02:42	Thi Ngo	Ready to Mark
2025/05/25 02:42	Mohammad Hossain	Time Exceeded
2025/05/25 02:49	Thi Ngo	Dear Dr. Iynkaran,Thank you for your constructive feedback on my initial report. I've taken the time to carefully revise the paper to address all of your comments in depth and illustrate the refinement part in blue color.Please find below summary of what I've worked on:- Focused Comparative Analysis: I developed a detailed case study comparing Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker. This includes a structured comparison table (Table 1) highlighting differences in data storage, training support, pipeline orchestration, and scalability.- Visual Contributions: I created a conceptual layered architecture diagram to illustrate how data storage layers interact with ML services, orchestration tools, and deployment systems. This was designed to reflect my own understanding of ML-DBMS integration workflows.- Critical Analysis: I identified key challenges such as high data movement latency, fragmented tooling, and lack of model governance. I then proposed corresponding solutions like

Deakin University

Database Fundamentals

OnTrack Submission

Research Report and Presentation

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/26 20:21

Tutor:
Mohammad belayet Hossain

Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

The integration of machine learning (ML) with database management systems (DBMS) represents a pivotal advancement in data-driven technologies. As organizations increasingly rely on intelligent systems to process vast and diverse datasets, understanding how ML and DBMS can effectively interoperate is crucial for building scalable, efficient, and reliable applications. This task aims to investigate the current state of ML-DBMS integration, identify prevailing challenges such as latency, model lifecycle management, and privacy compliance, and evaluate emerging solutions including federated learning, AutoML, and hybrid architectures. By conducting a comparative analysis of leading platforms like Google BigQuery with Vertex AI, Databricks with MLflow, and Amazon Redshift with SageMaker, the research provides critical insights into their operational trade-offs and practical applicability. The task's significance lies in addressing both theoretical gaps and real-world deployment issues, thereby contributing to the development of more robust, scalable, and ethically sound ML-enabled data infrastructures. Ultimately, this work supports the ongoing evolution of ML and data management technologies, ensuring their alignment with enterprise needs and regulatory requirements.

May 26, 2025

Research Report

The Interconnected Relationship Between Machine Learning and Database Systems: A Comparative Platform Study and Design Perspective

Lecturer: Dr. Iynkaran Natgunanathan

2025

Database Fundamentals - SIT772

Table of Content

1. Abstract

Using machine learning, systems learn by identifying patterns in data and improving performance on tasks without being explicitly programmed (Russell and Norvig, 2021). Machine learning is now fundamental in web searches, checking for fraud, giving credit scores and giving personalized advice and offers a solution that can be easily scaled over large numbers (Domingos, 2012). Still, achieving great results with ML models depends equally on good data and infrastructure as on how the models are built. A big part of practical ML is knowing about system topics such as how data should be collected, cleaned and stored (Domingos, 2012). With modern machine learning, relational databases can no longer keep up, making NoSQL databases, data lakes and MLflow tools attractive as they offer extensive scaling and pipeline automation (Stonebraker and Cetintemel, 2005; Zaharia et al., 2016). In addition, problems with data movement, proper model versioning and resource usage have made it difficult to connect ML and DBMS systems smoothly.

This article studies the way machine learning and data management are tied together and what current limits followed by emerging answers to these issues. It looks closely at combining Google BigQuery with Vertex AI and compares it to Amazon SageMaker and Databricks operated via MLflow, noticing the pros and cons each has for businesses. Furthermore, it suggests federated learning, AutoML and hybrid systems as answers to overcome privacy issues, enhance model performance and guarantee efficient use of resources in integrated systems. I believe that both ML and data infrastructure should improve together, as future progress will depend on new storage, privacy and modifiable ML solutions that fit with current database technologies.

2. Introduction

2.1 Overview of Machine Learning

A branch of artificial intelligence known as machine learning (ML) enables systems to recognise patterns and decide on actions no direct programming (Russell and Norvig, 2021). Over the years, ML has served many purposes in applications such as suggesting what to buy, finding fraud and processing language. The classification of ML algorithms is usually done between supervised, unsupervised and reinforcement learning. In supervised learning, data is pre-marked to help the model make predictions, but unsupervised learning looks for trends in

data that do not include labels (Russell and Norvig, 2021). Agents can learn the best way to act thanks to feedback given as rewards or penalties in reinforcement learning (Sutton and Barto, 2018). Modern statistical learning methods such as regression, classification and dimensionality reduction (Hastie, Tibshirani and Friedman, 2009) support these methods and form the foundation of most ML algorithms. As a result, such systems can improve themselves as they gather more information.

2.2 Importance of Data in ML

High-quality data in a good structure is necessary for ML systems to succeed. In the words of Domingos (2012), advanced algorithms make mistakes when their data is noisy, incomplete or not well labelled- "garbage in, garbage out". Managing a vast amount of data is not easy. They are well-suited to organised data, but are unable to handle the data traffic, variability and fast pace needed by today's ML pipelines. As Stonebraker and Cetintemel (2005) have pointed out, storing all data alike is no longer a good solution, which has led to greater use of NoSQL databases and data lakes for handling ML data. As ML systems gain more complexity, MLflow helps streamline the progress of ML from data preparation to deployment, by merging with contemporary data systems (Zaharia et al., 2016). ML models might not work well for long because data gradually changes which is called concept drift. The study indicates that regularly monitoring, keeping records of data changes and instructing your models all play key roles in handling a dynamic setting. Building ML systems that can adjust and grow requires careful design and control of the database.

3. Key Foundations Supporting The Integration Of Machine Learning And Data Systems

3.1 Data Requirements for ML

The data requirements for machine learning systems vary significantly based on the learning paradigm employed. Supervised learning typically demands large volumes of accurately labeled and well-structured datasets to establish reliable input-output mappings. Conversely, unsupervised learning operates on unlabeled data, often leveraging complex relationships in unstructured or semi-structured datasets to identify hidden patterns. Reinforcement learning involves sequential and time-dependent data obtained from agent-environment interactions, necessitating rich feedback mechanisms. These diverse

requirements necessitate careful consideration of not only data volume but also data format, provenance, and alignment with specific learning objectives.

3.1.1 Structured vs. Unstructured Data

A fundamental aspect of designing effective ML data pipelines lies in distinguishing between structured and unstructured data. Structured data is organized into predefined formats such as tabular schemas, facilitating storage and querying via relational databases. Examples include numeric test scores, identifiers, and timestamps commonly used in supervised learning contexts (Katal, Wazid, & Goudar, 2013). In contrast, unstructured data encompasses formats without fixed schema constraints, such as free text, audio, video, and open-ended survey responses, which are increasingly prevalent in modern ML applications like natural language processing and image recognition (Hashem et al., 2015). Semi-structured data, typified by JSON or XML, occupies an intermediate space with partial organization yet flexible formats (Russom, 2011). The complexity and heterogeneity of unstructured data impose significant challenges on preprocessing and feature extraction pipelines.

Successful ML systems thus rely on hybrid architectures capable of ingesting, storing, and processing both structured and unstructured data efficiently. This capability is critical in domains such as education and intelligent assistants, where input data ranges from formal records to informal text or multimedia content. Appendix C, Figure 1 (adapted from MongoDB), visually contrasts the key characteristics, storage methodologies, and application contexts of these data types (MongoDB, n.d.).

3.1.2 Volume, Variety, Velocity

The foundational characteristics of Big Data- volume, variety, and velocity continue to shape the data management challenges in ML applications (Laney, 2001). Volume refers to the extensive quantities of data necessary for training robust models, particularly deep learning architectures which often require millions of samples to achieve generalizability (Chen, Chiang, & Storey, 2012). Variety captures the diverse data modalities- ranging from structured tables to multimedia formats- that models must effectively process (Kwon, Lee, & Shin, 2014). Velocity denotes the speed of data generation and ingestion, crucial for real-time ML systems such as fraud detection or autonomous agents that must adapt instantaneously to new inputs (Zhang, Cao, & Muccini, 2020). The interplay among these dimensions informs data engineering

strategies, influencing model scalability, responsiveness, and overall system performance. Appendix D, Figure 2 illustrates the synergy of these factors in real-time, adaptive ML systems (Austin & Kusumoto, 2016).

3.1.3 The Importance of Data Quality

Data quality fundamentally affects ML model accuracy, fairness, and reliability. ML algorithms are especially vulnerable to issues such as noisy labels, missing data, class imbalances, and inconsistent formatting, which degrade performance and can lead to biased outcomes (Mohammed et al., 2022). High-quality datasets enable models to learn meaningful representations, reduce overfitting, and produce trustworthy predictions, which is paramount in high-stakes sectors including healthcare, finance, and autonomous systems (Zhou et al., 2024). Empirical evidence suggests that investing in data quality improvements often yields greater performance gains than increasing dataset size alone, making it a cost-effective enhancement strategy (Chen, 2022). Consequently, comprehensive data quality management is indispensable for responsible and scalable ML development.

3.2 Database and Storage Architectures for ML

Modern machine learning projects put a lot of pressure on systems storing and processing data, because of the large amount, widespread types and fast flow of data recognized as the 3Vs of big data (Laney, 2001). While traditional relational databases ensure good performance, they are frequently unable to meet the requirements for handling very large and mixed machine learning datasets (Moniruzzaman & Hossain, 2013; Khan et al., 2022). Thus, such systems are now built on large data platforms like distributed file systems and NoSQL databases which are equipped to process various kinds of data across many servers (Abuqabita et al., 2019; Cattell, 2011). Still, using machine learning tools in these systems often results in some concerns about operation. Easy-to-manage and automatically scalable serverless architectures usually make it difficult to change and control the steps in the pipeline (Sato et al., 2021). Alternatively, systems based on clusters allow for more flexibility and can be customized but add more difficulty to running them (Zaharia et al., 2016). Such concepts have led to adaptive hybrid systems that try to bring together efficient scaling, individual customization and manageability (Hashem et al., 2015; Albrecht et al., 2021).

In addition, data pipelines for machine learning are shifting from traditional ETL methods to ELT that handles data in the cloud to increase both efficiency and flexibility (Glickman & Rifkin, 2020; Breck et al., 2017). As batch and real-time data streams require more handling, it becomes important to use API integration and arrange data processing as separated modules that help freely transfer data in the system (Kreps et al., 2011; Li et al., 2020). These developments highlight the problem of balancing the needs of machine learning with data systems and show why both trends in machine learning and database technology are needed to achieve reliability and efficiency in future systems (Stonebraker & Cetintemel, 2005; Kraska et al., 2018).

3.3 Integrating ML Models with Databases

Since machine learning is being widely used in both enterprise and real-time applications, connecting ML models with databases directly becomes very important in system architecture. With old methods, ML training and deployment are handled apart from the main data systems, resulting in delays and extra costs for moving and changing data frequently. The use of ML inside database systems is being promoted in recent research because it allows for faster analytics and better consistency when moving from training to serving data (Kraska et al., 2024). Because of this shift, databases now actively contribute to ML tasks, allowing for the use of feature stores, registries for ML models and monitoring pipelines that make ML deployment more scalable (AWS, 2025).

Today's SQL and NoSQL databases are becoming better suited for ML tasks by offering features that let users perform predictions and create their customised functions, blending data management with ML (Chen, Zhang & Li, 2021). Thanks to ML, systems based on learnable databases can automatically improve indexing and query planning, prompting changes in the ways databases are built. Still, these integrations can create issues with resource allocation, different versions of models and understanding the system's decisions, meaning special attention is needed when designing the system for use in production. It proves that meeting ML needs and managing data sustainably is challenging which calls for the development of algorithms and database technologies together. Figure 6 in Appendix E makes clear that databases are key to the ML lifecycle, aiding in the smooth deployment of models by helping manage feature stores, models and their monitoring (AWS, 2025).

3.5 Limitations, Challenges, and Threats

While machine learning and database management systems have made great progress, significant challenges still prevent their wider use. Real-time inference is limited in finance and other sectors that demand it, because of the delays caused by regular communication between ML libraries and databases (Kraska et al., 2024; Gama et al., 2014). In addition, managing and checking the reproducibility of ML models is a problem since industry-wide standards for versioning models have not been set (Abadi et al., 2016; Zaharia et al., 2018).

Running both ML tasks and regular DBMS workloads together can affect the system's dependability and performance, according to Biondi et al. (2023). Using different tools and APIs creates difficulties for the proper use of DBMS in companies, leads to the complication of deployment and lifecycle operations and reduces chances for automation (Chen, Zhang & Li, 2021).

Various new strategies are promising for dealing with these matters.

- With federated learning, each endpoint trains the algorithm with its local data, avoiding data sharing and respecting privacy, as required by regulations such as the GDPR, says Kairouz et al. in 2019.
- AutoML frameworks allow anyone to implement Machine Learning because they automate important tasks, do not need experts and ensure results can be reproduced (He et al., 2021).
- Having model registries and unified APIs for controlling versions and deployment can improve governance and how operations are run (According to Abadi et al., 2016).
- Mixing in-database machine learning with external methods can improve performance and make better use of machine learning resources (Chen, Zhang & Li, 2021).

This is important as we seek to build ML systems that are secure, can scale and remain in compliance. Key issues with merging ML and DBMS are shown in Table 2, along with the latest solutions suggested by research.

Identified Challenge	Implications	Proposed Solution & Citations
High latency due to data movement	Limits real-time inference; slows response	Federated learning to reduce data transfer (Kairouz et al., 2019)
Lack of model versioning standards	Hinders reproducibility and auditing	Standardised model registries and APIs (Abadi et al., 2016)
Resource contention between DB and ML workloads	Degraded performance and reliability	Hybrid architectures separating inference and training (Chen, Zhang & Li, 2021)
Fragmented tooling ecosystem	Operational complexity, reduced automation	Unified ML lifecycle platforms and integrated tooling (Zaharia et al., 2018)
Privacy and compliance challenges	Risk of regulatory violations and bias	Privacy-preserving computation embedded in DBMS (Barocas et al., 2019; Kairouz et al., 2019)

Table 1: Problem-Solution Matrix summarises the key limitations in ML-DBMS integration alongside proposed solutions

3.5.1 Ethical and Privacy Challenges

The integration of ML-DBMS poses several big challenges around ethics and privacy, too. Concerns such as unequal use of algorithms, abuses of data and a lack of transparency risk both a user's confidence and the fairness of the system. Following GDPR means privacy practices are built directly into the process of using databases and ML algorithms. Proper AI use is achieved by focusing on differential privacy, explainable AI and audit trails (Barocas et al., 2019; Kairouz et al., 2019).

3.6 Solutions and Tools from Industry

Industry tools such as Google's Vertex AI paired with BigQuery, Databricks with MLflow and AWS Redshift with SageMaker, solve many of the problems with integrating ML and DBMS.

These platforms help manage every stage of machine learning, including working with data, making models and using them in applications, dealing with problems such as latency and resource usage. Nevertheless, it is still difficult to achieve standardisation, but Federated Learning and AutoML promise to help address privacy, accessibility and performance concerns (Kairouz et al., 2019; He et al., 2021).

3.7 Comparative Analysis of ML-DBMS Platforms

Despite significant progress in ML-DBMS integration, practical deployment challenges and research gaps remain. For example, deploying ML models in regulated environments demands strict compliance with privacy laws, yet current architectures often lack built-in auditing and protection features. Moreover, the heterogeneity of data types and ML workloads complicates the design of universal, scalable pipelines. Practitioners struggle to balance complexity and performance, a tension insufficiently addressed in existing research. Addressing these gaps is essential for bridging theory and practice in scalable ML-DBMS systems.

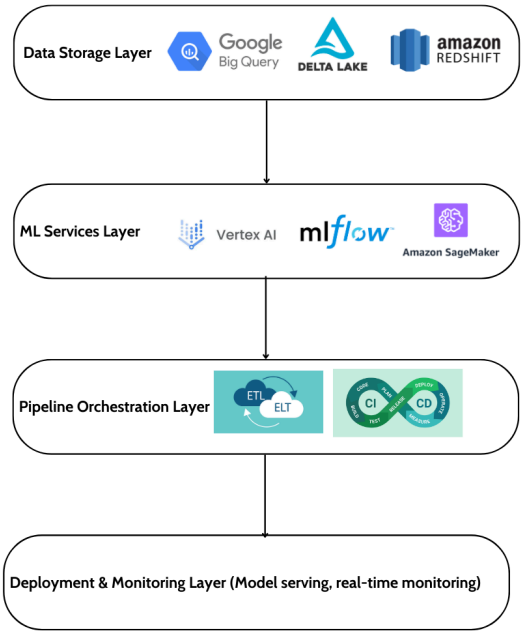
The integration of ML capabilities into database management systems has advanced through platforms such as Google BigQuery with Vertex AI, Databricks paired with MLflow, and Amazon Redshift combined with SageMaker. These platforms demonstrate distinct approaches to data storage, model lifecycle management, scalability, and pipeline orchestration.

Feature	Google BigQuery + Vertex AI	Databricks + MLflow	Amazon Redshift + SageMaker
Data Storage Type	Serverless columnar warehouse	Lakehouse architecture with Delta Lake	Cloud data warehouse in the AWS ecosystem
ML Model Training Support	Integrated AutoML and custom model training	Experiment tracking, model registry, Spark-based	Managed notebooks and training with SageMaker
Deployment & Monitoring	Pipeline automation via Vertex AI Pipelines	Model reproducibility and multi-platform deployment	Real-time model monitoring with SageMaker
Scalability	Fully serverless, automatic scaling	Scalable Spark clusters requiring management	Horizontally scalable, deep AWS integration
Pipeline Integration	Tight ELT integration with BigQuery and Vertex AI	Notebook-driven workflows with CI/CD support	Integration with AWS Step Functions and Lambda
Limitations & Challenges	Complex pricing, learning curve	Cluster management overhead	Vendor lock-in, operational cost concerns

Table 2: Feature comparison of leading ML-integrated DBMS platforms based on official documentation and academic studies (Sato et al., 2021; Zaharia et al., 2018; AWS, 2024).

Using Google BigQuery and Vertex AI means the system can handle any size of dataset without servers. As a result, this model cuts down on running costs and allows models to be deployed quickly. On the other hand, because DataFlow has fewer customization options in its pipelines, it works best in simple cases and can still be deployed more quickly. But if you have advanced expertise, Databricks with MLflow gives you more flexible options to work with. Using Spark clusters, Databricks assures you have sufficient bandwidth to run big, scattered datasets. Handling the management of Spark clusters can be difficult for companies that have limited resources and support. With Redshift from Amazon and SageMaker combined on AWS, you can store your data, train ML models and deploy them in a straightfoward way. Scaling with

Redshift is reliable, but its strong connection to AWS could cause problems for companies as their workload increases, since this becomes expensive. This comparison makes it clear that sometimes it's simple to manage, but not as flexible and vice versa. Because BigQuery is serverless, there is less to operate, but you have less control over your pipelines. While Databricks lets you customize your solution, it needs careful cluster management, whereas using AWS gives you easier integration and may require more effort to switch to other vendors (Sato et al., 2021; Zaharia et al., 2018; AWS, 2024).



This diagram illustrates a layered architecture where the data storage layer (DBMS) interfaces directly with integrated ML services (e.g., Vertex AI, MLflow, SageMaker). This diagram illustrates four interdependent layers: the Data Storage Layer, where platforms like BigQuery, Delta Lake, and Redshift reside; the ML Services Layer, containing tools such as Vertex AI, MLflow, and SageMaker; the Pipeline Orchestration Layer, which manages data workflows (e.g., ELT) and CI/CD processes; and the Deployment & Monitoring Layer, responsible for model serving and real-time tracking. The architecture shows a sequential data flow from ingestion to inference, reinforced by feedback loops for continuous learning and lifecycle monitoring.This layered model emphasizes the necessity of tight integration between

data management systems and machine learning services to achieve low-latency inference, robust governance, and efficient resource utilization. It further illustrates how architectural decisions at each layer affect scalability, flexibility, and operational complexity- key factors analyzed throughout this study.

4. Future Directions

Developments in both machine learning (ML) and data management are accelerating in response to new developments that emphasize privacy, accessibility and keeping operations consistent. Another trend called federated learning addresses these challenges by training models on data kept by different parties without the need to transfer their raw data (Kairouz et al., 2019). Federated learning helps lessen need for big data storage by having every participant add their updated model only. Sectors such as healthcare and finance gain the most from this technique since protecting data is very important.

By automating crucial steps in the machine learning pipeline, including as feature engineering, model selection, and hyperparameter tuning, another movement called Automated Machine Learning (AutoML) is changing accessibility (He et al., 2021). By democratizing machine learning, non-experts may now create competitive models, reducing entry barriers and spurring innovation in a range of sectors. However, issues with standardization, reproducibility, model governance, and scalability continue to arise as machine learning is incorporated more and more into intricate database systems. Widespread adoption is hampered by the complexity of integrating machine learning with data systems due to the absence of standardized frameworks.

To address these issues, there is an urgent need for standardized frameworks that can harmonize data pipelines, training procedures, and deployment processes across different platforms (Abadi et al., 2016). Additionally, cloud-native architectures, which integrate ML services directly into the database management layer, can enhance real-time data processing and reduce latency in data-intensive applications.These advancements suggest a future where ML systems are not only more inclusive and performant but also more secure, transparent, and dependable, enabling organizations to adopt ML more confidently while maintaining high ethical and operational standards.

5. Conclusion

In conclusion, the integration of machine learning with database technologies is pivotal to the development of modern data-driven systems. Successful ML applications depend not only on sophisticated algorithms but also on well-managed, high-quality data. As ML and database infrastructures evolve together, solutions like federated learning, AutoML, and hybrid architectures show great potential for overcoming challenges such as privacy, scalability, and resource contention. Moving forward, standardization and improved cloud-native architectures will be key to enabling scalable, reproducible, and ethical ML-DBMS systems.

6. References

Austin, C. and Kusumoto, F. (2016). *The application of Big Data in medicine: current implications and future directions*. Journal of Interventional Cardiac Electrophysiology, 47(1), pp.51–59. https://doi.org/10.1007/s10840-016-0104-y

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp.265–283.

Amazon Web Services (AWS). (2025). *Machine Learning Lens: ML lifecycle architecture diagram*. Available at: https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/ml-lifecycle-architecture-diagram.html [Accessed 5 April 2025].

Abuqabita, F., Al-Omoush, R. and Alwidian, J. (2019). *A comparative study of big data analytics frameworks*. International Journal of Advanced Computer Science and Applications, 10(5), pp.370–377. https://doi.org/10.14569/IJACSA.2019.0100548

Amazon Web Services (AWS). (2024). *The difference between ETL and ELT*. Available at: https://aws.amazon.com/compare/the-difference-between-etl-and-elt/ [Accessed 10 April 2025].

Biondi, F., Pinto, G., Sá, C., Capiluppi, A., & D'Amico, G. (2023). *Understanding the challenges of architecting ML-enabled systems: A multi-method study*. Journal of Systems and Software, 200, 111645. https://doi.org/10.1016/j.jss.2023.111645

Barocas, S., Hardt, M. and Narayanan, A., 2019. *Fairness and Machine Learning*. Cambridge, MA: fairmlbook.org. [online] Available at: https://fairmlbook.org/ [Accessed 28 Apr. 2025].

Breck, E., Polyzotis, N., Roy, S., Whang, S.E. and Zinkevich, M. (2017). *Data validation for machine learning*. Proceedings of SysML Conference. Available at: https://mlsys.org/Conferences/doc/2018/9.pdf [Accessed 20 April 2025].

Chen, H., Chiang, R.H.L. and Storey, V.C. (2012). *Business Intelligence and Analytics: From Big Data to Big Impact*. MIS Quarterly, 36(4), pp.1165–1188. https://doi.org/10.2307/41703503

Chen, H. (2022). *Data quality for artificial intelligence*. SSRN. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4280581 [Accessed 20 April 2025].

Chen, L., Zhang, Y. and Li, J. (2021). *Survey on Learnable Databases: A Machine Learning Perspective*. Big Data Research, 25, 100215. https://doi.org/10.1016/j.bdr.2021.100215

Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM*, 55(10), pp.78–87. https://doi.org/10.1145/2347736.2347755

Gama, J., Zliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A., 2014. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4), pp.1–37. https://doi.org/10.1145/2523813

Glickman, S. and Rifkin, A., 2020. The future of data engineering: ELT, not ETL. *Towards Data Science*. Available at: https://towardsdatascience.com/the-future-of-data-engineering-elt-not-etl-2e59a7b4c26e [Accessed 28 Apr. 2025].

Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer. Available at: https://web.stanford.edu/~hastie/ElemStatLearn/ [Accessed 4 May 2025].

He, X., Zhao, K. and Chu, X., 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, p.106622. https://doi.org/10.1016/j.knosys.2020.106622

Katal, A., Wazid, M. and Goudar, R.H., 2013. Big data: Issues, challenges, tools and good practices. In: *2013 Sixth International Conference on Contemporary Computing (IC3)*, pp.404–409. https://doi.org/10.1109/IC3.2013.6612229

Kwon, O., Lee, N. and Shin, B. (2014). *Data quality management, data usage experience and acquisition intention of big data analytics*. International Journal of Information Management, 34(3), pp.387–394. https://doi.org/10.1016/j.ijinfomgt.2014.02.002

Khan, W., Kumar, T., Cheng, Z., Raj, K., Roy, A.M. and Luo, B. (2022). *Performance analysis of SQL and NoSQL databases for big data applications: A systematic literature review*. Computers, Materials & Continua, 70(2), pp.3343–3363. https://doi.org/10.32604/cmc.2022.020203

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R. and D'Oliveira, R.G.L., 2019. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2), pp.1–210. https://doi.org/10.1561/2200000083

Kreps, J., 2011. The Log: What every software engineer should know about real-time data's unifying abstraction. *LinkedIn Engineering Blog*. Available at: https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-data [Accessed 28 Apr. 2025].

MongoDB, n.d. Structured vs. Unstructured Data. [online] Available at: https://www.mongodb.com/resources/basics/unstructured-data/structured-vs-unstructured [Accessed 5 April 2025].

Mohammed, S., Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F. and Harmouch, H. (2022). *On the impact of data quality on machine learning: A survey*. arXiv preprint arXiv:2207.14529. Available at: https://arxiv.org/abs/2207.14529 [Accessed 22 April 2025].

Moniruzzaman, A.B.M. (2014). *NewSQL: Towards next-generation scalable RDBMS for online transaction processing (OLTP) for big data management*. arXiv preprint arXiv:1406.5061. Available at: https://arxiv.org/abs/1406.5061 [Accessed 26 April 2025].

Sutton, R.S. and Barto, A.G., 2018. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press. Available at: http://incompleteideas.net/book/the-book.html [Accessed 4 May 2025].

Stonebraker, M. and Çetintemel, U., 2005. One size fits all: An idea whose time has come and gone. In: *21st International Conference on Data Engineering (ICDE)*. Tokyo: IEEE, pp.2–11. https://doi.org/10.1109/ICDE.2005.1

Sato, H., Imai, M. and Minami, Y., 2021. Serverless Data Analytics with BigQuery: Challenges and Solutions. *Proceedings of the VLDB Endowment*, 14(12), pp.3197–3209. https://doi.org/10.14778/3476311.3476380

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Park, J., Sen, R. and Xin, R., 2016. Accelerating the machine learning lifecycle with MLflow. *Databricks*. Available at: https://www.mlflow.org/docs/latest/index.html [Accessed 28 Apr. 2025].

Zhang, P., Cao, W. and Muccini, H. (2020). *Quality assurance technologies for big data applications: A systematic literature review*. Journal of Systems and Software, 161, p.110463. https://doi.org/10.1016/j.jss.2019.110463

Zhou, Y., Tu, F., Sha, K., Ding, J. and Chen, H. (2024). *Data quality for machine learning: A survey*. arXiv preprint arXiv:2406.19614. Available at: https://arxiv.org/pdf/2406.19614 [Accessed 22 April 2025].

Zhao, M., Lee, Y., Xie, J., Ouyang, P., Ding, Y., Zhou, Y., & Yu, F. (2021). *Storage and ingestion for large-scale deep recommendation model training*. arXiv preprint arXiv:2108.09373. Available at: https://arxiv.org/abs/2108.09373 [Accessed 28 April 2025].

Zhang, Y., Liu, H. and Wang, X. (2024). *Adaptive and Scalable Database Management with Machine Learning*. Information, 15(9), 574. https://doi.org/10.3390/info15090574

Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S. and Stoica, I., 2018. Apache Spark: A unified engine for big data processing. *Communications of the ACM*, 59(11), pp.56–65. https://doi.org/10.1145/2934664

Russell, S. and Norvig, P., 2021. *Artificial Intelligence: A Modern Approach*. 4th ed. London: Pearson.

Russom, P., 2011. *Big Data Analytics*. TDWI Best Practices Report. Available at: https://tdwi.org/research/2011/09/best-practices-report-q4-big-data-analytics.aspx [Accessed 4 May 2025].

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Khan, S.U., 2015. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47, pp.98–115. https://doi.org/10.1016/j.is.2014.07.006

Laney, D. (2001). *3D data management: Controlling data volume, velocity and variety*. META Group Research Note, 6(70), pp.1–4.

Li, M., Zeng, X., Jiang, Z., Zhang, B., Shi, Z., Zhang, C., & Ren, X. (2024). *Social AI Data Infrastructure: A data library and taxonomy for social intelligence in NLP*. arXiv preprint arXiv:2403.14659. Available at: https://arxiv.org/abs/2403.14659 [Accessed 20 April 2025].

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J. and Zhang, G., 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12), pp.2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

Kraska, T., Beutel, A., Chi, E.H., Dean, J. and Polyzotis, N. (2024). *Serving Deep Learning Models from Relational Databases*. In: Proceedings of the International Conference on Extending Database Technology (EDBT), pp.174–185. https://doi.org/10.5441/002/edbt.2024.15

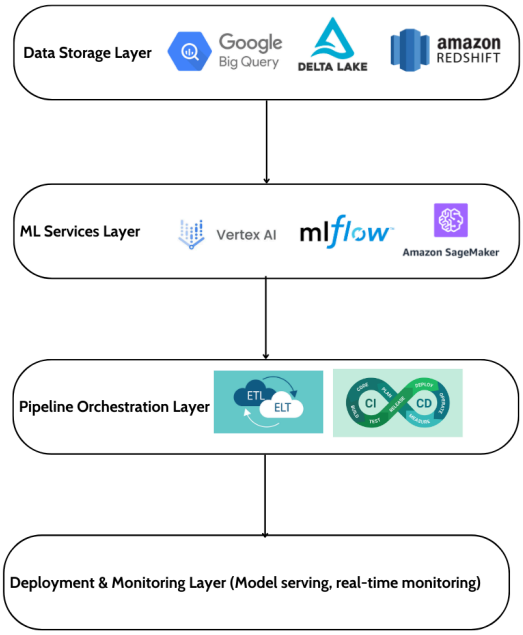
Piech, C., Sahami, M., Koller, D., Cooper, S., Blikstein, P. and Xu, Y., 2015. Modeling how students learn to program. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*, pp. 313–318. https://doi.org/10.1145/2676723.2677252

7. Appendice

Appendix A: Relevance to Coursera

Coursera, as one of the world's leading online learning platforms, heavily relies on machine learning (ML) and robust data management systems to enhance its services and support its global user base. Through the use of ML algorithms, Coursera personalizes course recommendations, predicts learner engagement, and identifies students at risk of dropping out, allowing for timely interventions that improve retention and learning outcomes (Piech et al., 2015). Databases play a critical role in storing and managing massive volumes of learner activity data, assessment records, and interaction logs, which serve as the foundation for training and updating these ML models. As Coursera continues to grow, opportunities arise to further leverage predictive analytics, adaptive learning pathways, and intelligent support systems to create even more personalized and effective educational experiences. However, challenges such as maintaining user privacy, adhering to international data protection regulations like GDPR, and managing the complexity of large-scale, multi-region data infrastructures remain significant (Kairouz et al., 2019). Additionally, ensuring fairness and avoiding algorithmic biases are crucial as the platform seeks to provide equitable access to learners from diverse backgrounds. Addressing these challenges while advancing ML capabilities will be vital for Coursera to sustain its competitive edge and continue scaling in the evolving ed-tech landscape.

Appendix B: Conceptual Architecture of ML-Integrated DBMS Platforms



This diagram illustrates a layered architecture where the data storage layer (DBMS) interfaces directly with integrated ML services (e.g., Vertex AI, MLflow, SageMaker). It highlights the pipeline orchestration layer managing ELT workflows and CI/CD pipelines, and a deployment & monitoring layer overseeing automated model serving and real-time performance tracking. Data flows sequentially from ingestion to inference and feedback loops enable continuous learning.

This conceptual model emphasizes the tight coupling required between data and ML services to minimize latency and improve lifecycle management.

Appendix C: Comparison between Structured and Unstructured Data.

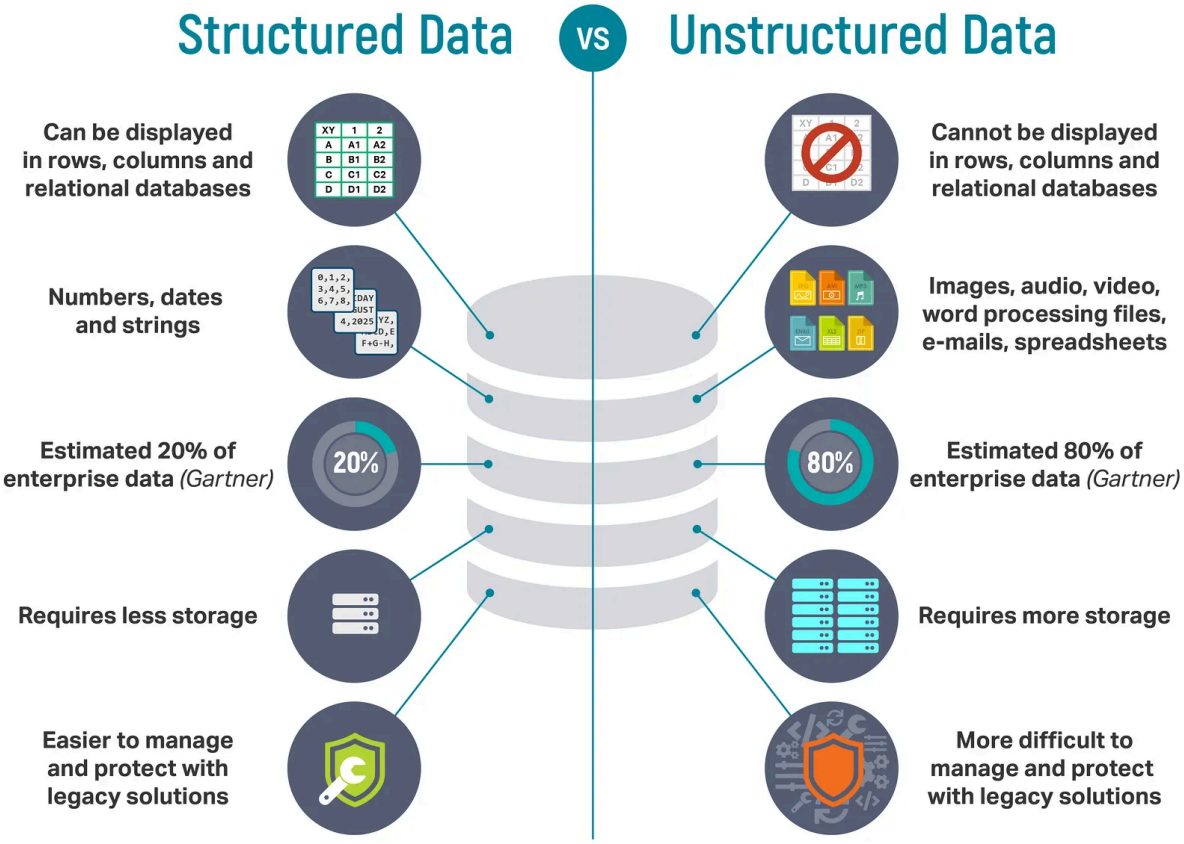


Figure 1: Comparison between Structured and Unstructured Data. Source: MongoDB.

Appendix D: The 3 Vs of Big Data.

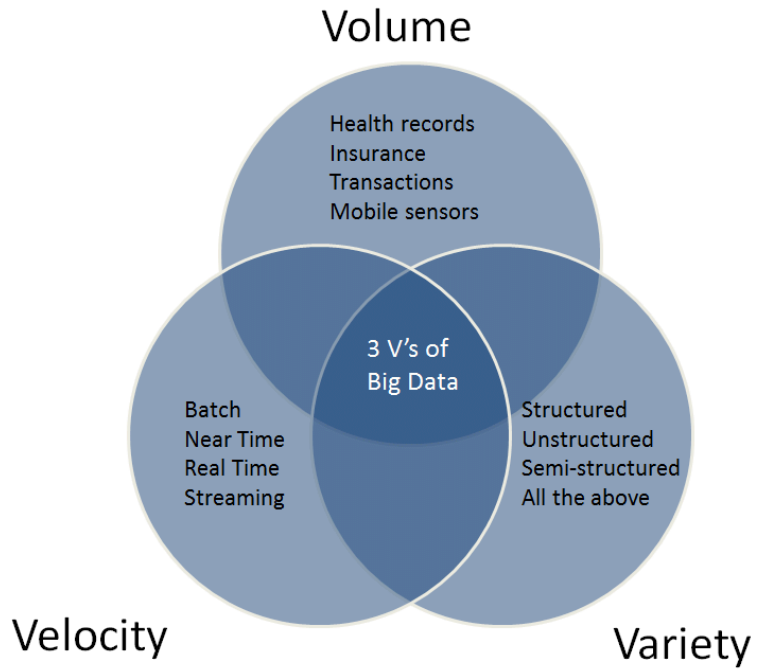


Figure 2: The 3 Vs of Big Data. Volume, variety and velocity have been widely accepted as the key data management challenges associated with Big Data

Appendix E - Machine Learning Lifecycle Architecture Diagram

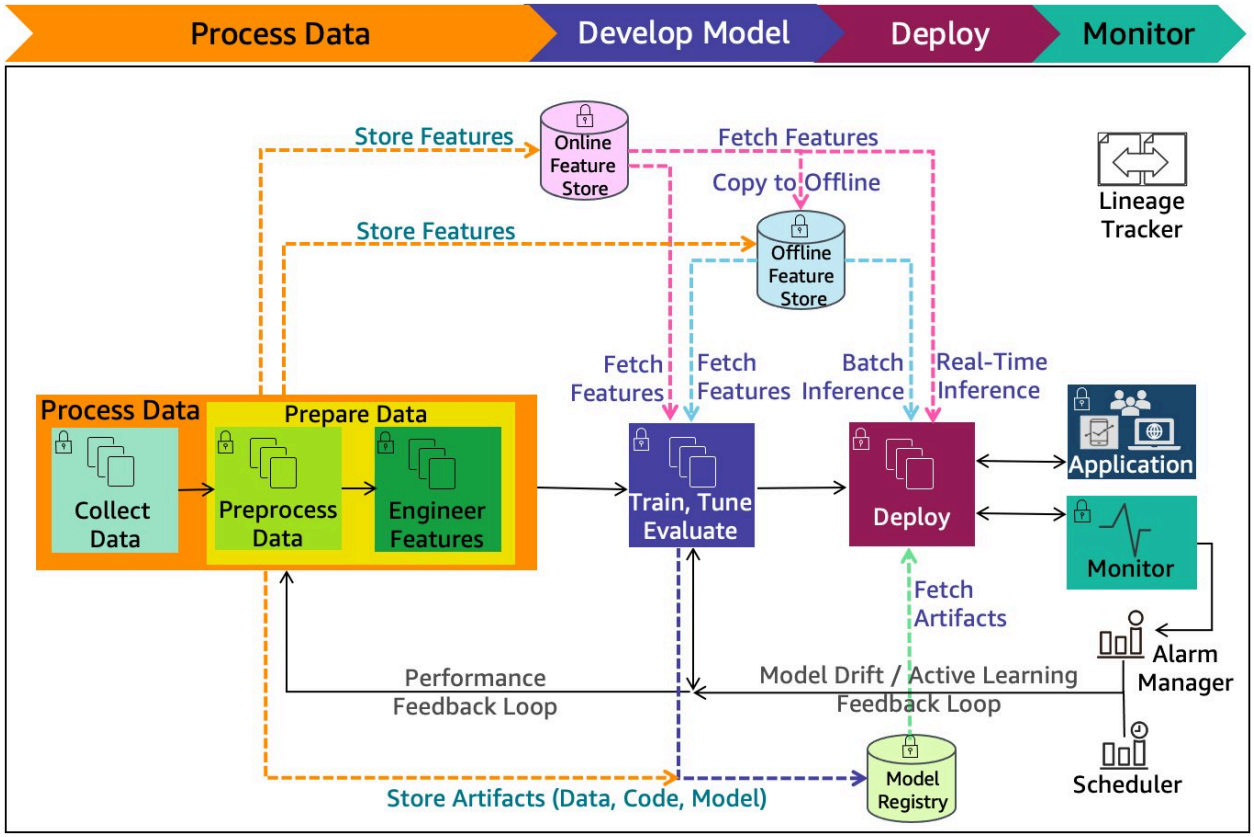


Figure 6: Machine Learning Lifecycle Architecture Diagram

15 Miniproject Part-2 - Database Implementation

Implement and manage database designed in Task 4.2C

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

Outcome	Weight
Reflection	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

Date	Author	Comment
2025/04/17 14:35	Thi Ngo	Working On It
2025/04/17 14:35	Thi Ngo	Not Started
2025/04/27 22:54	Thi Ngo	Working On It
2025/05/07 16:11	Thi Ngo	Ready to Mark
2025/05/08 13:27	Mohammad Hossain	Good work.
2025/05/08 13:27	Mohammad Hossain	Complete
2025/05/19 17:46	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Miniproject Part-2 - Database Implementation

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/07 16:11

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through completing Task 7.2C, I learned how to implement a normalized database schema using SQL by creating tables with primary and foreign keys, inserting and managing data, and modifying existing structures with ALTER and UPDATE statements. I practiced using SQL queries to retrieve and manipulate data through JOIN, WHERE, IN, and DATE functions, as well as how to create and use database views. This task strengthened my understanding of relational integrity, query logic, and the importance of structuring data for efficient access and reporting.

May 7, 2025

1) Write SQL statements to create all tables in the final normalized ERD in Task 4.2C

```
CREATE TABLE INSTITUTION (
    INSTITUTION_ID INT PRIMARY KEY,
    NAME VARCHAR(100) NOT NULL,
    COUNTRY VARCHAR(50),
    WEBSITE VARCHAR(255)
);

CREATE TABLE INSTRUCTOR (
    INSTRUCTOR_ID INT PRIMARY KEY,
    NAME VARCHAR(100) NOT NULL,
    EMAIL VARCHAR(100) UNIQUE,
    INSTITUTION_ID INT,
    FOREIGN KEY (INSTITUTION_ID) REFERENCES INSTITUTION(INSTITUTION_ID)
);

CREATE TABLE USER (
    USER_ID INT PRIMARY KEY,
    NAME VARCHAR(100) NOT NULL,
    EMAIL VARCHAR(100) UNIQUE NOT NULL,
    PASSWORD VARCHAR(100) NOT NULL,
    COUNTRY VARCHAR(50),
    JOIN_DATE DATE
);

CREATE TABLE COURSE (
    COURSE_ID INT PRIMARY KEY,
    TITLE VARCHAR(200) NOT NULL,
    DESCRIPTION TEXT,
    CATEGORY VARCHAR(50),
    INSTITUTION_ID INT,
    INSTRUCTOR_ID INT,
    FOREIGN KEY (INSTITUTION_ID) REFERENCES INSTITUTION(INSTITUTION_ID),
    FOREIGN KEY (INSTRUCTOR_ID) REFERENCES INSTRUCTOR(INSTRUCTOR_ID)
);

CREATE TABLE ENROLLMENT (
    ENROLLMENT_ID INT PRIMARY KEY,
    USER_ID INT,
    COURSE_ID INT,
    STATUS VARCHAR(50),
```

```
    CERTIFICATE_ISSUED BOOLEAN,
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID),
    FOREIGN KEY (COURSE_ID) REFERENCES COURSE(COURSE_ID)
);

CREATE TABLE CERTIFICATE (
    CERTIFICATE_ID INT PRIMARY KEY,
    ENROLLMENT_ID INT UNIQUE,
    ISSUE_DATE DATE,
    CERTIFICATE_URL VARCHAR(255),
    FOREIGN KEY (ENROLLMENT_ID) REFERENCES ENROLLMENT(ENROLLMENT_ID)
);

CREATE TABLE ASSIGNMENT (
    ASSIGNMENT_ID INT PRIMARY KEY,
    COURSE_ID INT,
    TITLE VARCHAR(200),
    DUE_DATE DATE,
    MAX_SCORE FLOAT,
    FOREIGN KEY (COURSE_ID) REFERENCES COURSE(COURSE_ID)
);

CREATE TABLE SUBMISSION (
    SUBMISSION_ID INT PRIMARY KEY,
    ASSIGNMENT_ID INT,
    USER_ID INT,
    SUBMISSION_DATE DATE,
    FILE_URL VARCHAR(255),
    SCORE FLOAT,
    FOREIGN KEY (ASSIGNMENT_ID) REFERENCES ASSIGNMENT(ASSIGNMENT_ID),
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID)
);

CREATE TABLE SUBSCRIPTION (
    SUBSCRIPTION_ID INT PRIMARY KEY,
    USER_ID INT UNIQUE,
    PLAN_TYPE VARCHAR(50),
    START_DATE DATE,
    END_DATE DATE,
    AUTO_RENEW BOOLEAN,
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID)
);
```

2) Insert at least FIVE records in each table using SQL.

```
INSERT INTO INSTITUTION VALUES
(1, 'Stanford University', 'USA', 'https://stanford.edu'),
(2, 'University of Melbourne', 'Australia', 'https://unimelb.edu.au'),
(3, 'MIT', 'USA', 'https://mit.edu'),
(4, 'University of Tokyo', 'Japan', 'https://u-tokyo.ac.jp'),
(5, 'Oxford University', 'UK', 'https://ox.ac.uk');

INSERT INTO INSTRUCTOR VALUES
(101, 'Dr. Alice Smith', 'alice@stanford.edu', 1),
(102, 'Prof. John Lee', 'john@unimelb.edu.au', 2),
(103, 'Dr. Emma Brown', 'emma@mit.edu', 3),
(104, 'Dr. Satoshi Yamamoto', 'satoshi@u-tokyo.ac.jp', 4),
(105, 'Prof. Sarah Green', 'sarah@ox.ac.uk', 5);

INSERT INTO USER VALUES
(201, 'Nguyen Tuan', 'tuan@gmail.com', 'pass123', 'Vietnam', '2024-01-01'),
(202, 'James Walker', 'james@gmail.com', 'walker456', 'Australia', '2023-12-20'),
(203, 'Hiro Tanaka', 'hiro@gmail.com', 'tanaka789', 'Japan', '2023-11-15'),
(204, 'Lily Chen', 'lily@gmail.com', 'lilyabc', 'China', '2024-02-10'),
(205, 'Sarah Kim', 'sarahk@gmail.com', 'sarah321', 'South Korea', '2023-10-30');

INSERT INTO COURSE VALUES
(301, 'Machine Learning', 'Intro to ML', 'Data Science', 1, 101),
(302, 'AI Ethics', 'Understanding ethics in AI', 'AI', 2, 102),
(303, 'Web Development', 'HTML, CSS, JS basics', 'Programming', 3, 103),
(304, 'Robotics', 'Designing robotic systems', 'Engineering', 4, 104),
(305, 'Philosophy of Mind', 'Cognitive science', 'Philosophy', 5, 105);

INSERT INTO ENROLLMENT VALUES
(401, 201, 301, 'Active', TRUE),
(402, 202, 302, 'Completed', TRUE),
(403, 203, 303, 'Active', FALSE),
(404, 204, 304, 'Completed', TRUE),
(405, 205, 305, 'Dropped', FALSE),
(406, 201, 302, 'Completed', TRUE),
(407, 203, 301, 'Completed', TRUE);



INSERT INTO CERTIFICATE VALUES
(501, 401, '2024-03-01', 'https://coursera.org/cert/401'),
(502, 402, '2023-12-25', 'https://coursera.org/cert/402'),
```

(503, 404, '2024-04-15', 'https://coursera.org/cert/404'),
(504, 406, '2024-04-25', 'https://coursera.org/cert/406'),
(505, 407, '2024-04-27', 'https://coursera.org/cert/407');

INSERT INTO ASSIGNMENT VALUES
(601, 301, 'ML Quiz 1', '2024-03-10', 100),
(602, 302, 'Essay on Ethics', '2024-02-15', 50),
(603, 303, 'Build a Web Page', '2024-03-05', 75),
(604, 304, 'Robot Arm Design', '2024-04-01', 80),
(605, 305, 'Reflection Paper', '2024-02-28', 60);

INSERT INTO SUBMISSION VALUES
(701, 601, 201, '2024-03-05', 'ml1.pdf', 95),
(702, 602, 202, '2024-02-10', 'ethics.docx', 45),
(703, 603, 203, '2024-03-03', 'webpage.zip', 70),
(704, 604, 204, '2024-03-28', 'robot.png', 78),
(705, 605, 205, '2024-02-25', 'mind.pdf', 58);

INSERT INTO SUBSCRIPTION VALUES
(801, 201, 'Monthly', '2024-01-01', '2024-04-01', TRUE),
(802, 202, 'Annual', '2023-12-01', '2024-12-01', TRUE),
(803, 203, 'Free Trial', '2024-03-01', '2024-03-31', FALSE),
(804, 204, 'Monthly', '2024-02-01', '2024-05-01', TRUE),
(805, 205, 'Annual', '2023-11-15', '2024-11-15', TRUE);

SELECT * FROM INSTITUTION;

INSTITUTION_ID	NAME	COUNTRY	WEBSITE
1	Stanford University	USA	https://stanford.edu
2	University of Melbourne	Australia	https://unimelb.edu.au
3	MIT	USA	https://mit.edu
4	University of Tokyo	Japan	https://u-tokyo.ac.jp
5	Oxford University	UK	https://ox.ac.uk
NULL	NULL	NULL	NULL

SELECT * FROM INSTRUCTOR;

INSTRUCTOR_ID	NAME	EMAIL	INSTITUTION_ID
101	Dr. Alice Smith	alice@stanford.edu	1
102	Prof. John Lee	john@unimelb.edu.au	2
103	Dr. Emma Brown	emma@mit.edu	3
104	Dr. Satoshi Yamamoto	satoshi@u-tokyo.ac.jp	4
105	Prof. Sarah Green	sarah@ox.ac.uk	5
NULL	NULL	NULL	NULL

SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30
NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM COURSE;

COURSE_ID	TITLE	DESCRIPTION	CATEGORY	INSTITUTION_ID	INSTRUCTOR_ID
301	Machine Learning	Intro to ML	Data Science	1	101
302	AI Ethics	Understanding ethics in AI	AI	2	102
303	Web Development	HTML, CSS, JS basics	Programming	3	103
304	Robotics	Designing robotic systems	Engineering	4	104
305	Philosophy of Mind	Cognitive science	Philosophy	5	105
NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM ENROLLMENT;

ENROLLMENT_ID	USER_ID	COURSE_ID	STATUS	CERTIFICATE_ISSUED
401	201	301	Active	1
402	202	302	Completed	1
403	203	303	Active	0
404	204	304	Completed	1
405	205	305	Dropped	0
406	201	302	Completed	1
407	203	301	Completed	1
NULL	NULL	NULL	NULL	NULL

SELECT * FROM CERTIFICATE;

CERTIFICATE_ID	ENROLLMENT_ID	ISSUE_DATE	CERTIFICATE_URL
501	401	2024-03-01	https://coursera.org/cert/401
502	402	2023-12-25	https://coursera.org/cert/402
503	404	2024-04-15	https://coursera.org/cert/404
504	406	2024-04-25	https://coursera.org/cert/406
505	407	2024-04-27	https://coursera.org/cert/407
NULL	NULL	NULL	NULL

SELECT * FROM ASSIGNMENT;

ASSIGNMENT_ID	COURSE_ID	TITLE	DUE_DATE	MAX_SCORE
601	301	ML Quiz 1	2024-03-10	100
602	302	Essay on Ethics	2024-02-15	50
603	303	Build a Web Page	2024-03-05	75
604	304	Robot Arm Design	2024-04-01	80
605	305	Reflection Paper	2024-02-28	60
NULL	NULL	NULL	NULL	NULL

SELECT * FROM SUBMISSION;

SUBMISSION_ID	ASSIGNMENT_ID	USER_ID	SUBMISSION_DATE	FILE_URL	SCORE
701	601	201	2024-03-05	ml1.pdf	95
702	602	202	2024-02-10	ethics.docx	45
703	603	203	2024-03-03	webpage.zip	70
704	604	204	2024-03-28	robot.png	78
705	605	205	2024-02-25	mind.pdf	58
NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM SUBSCRIPTION;

SUBSCRIPTION_ID	USER_ID	PLAN_TYPE	START_DATE	END_DATE	AUTO_RENEW
801	201	Monthly	2024-01-01	2024-04-01	1
802	202	Annual	2023-12-01	2024-12-01	1
803	203	Free Trial	2024-03-01	2024-03-31	0
804	204	Monthly	2024-02-01	2024-05-01	1
805	205	Annual	2023-11-15	2024-11-15	1
NULL	NULL	NULL	NULL	NULL	NULL

3) Write and run an SQL statement to add one additional column (attribute) in any one of the existing tables with a default value.

- Add New Column `IS_ACTIVE` to USER Table
- **Description:** I added a new column `IS_ACTIVE` (BOOLEAN) to the `USER` table, with a default value of `TRUE`, to indicate if a user account is active.
- **SQL Statement:**

```
ALTER TABLE USER
ADD COLUMN IS_ACTIVE BOOLEAN DEFAULT TRUE;
```

- **Screenshot:**

SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE	IS_ACTIVE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01	1
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20	1
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15	1
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10	1
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4) Write and run an SQL statement to update the default value of the newly added column to a different value for certain rows based on a condition using any other column.

- Update USER.IS_ACTIVE to FALSE for users who have any enrollment with STATUS = 'Dropped'.
- **Description:** I updated the IS_ACTIVE column to FALSE for all users who have dropped a course. This was done by checking for users in the ENROLLMENT table where the STATUS is 'Dropped', then applying the change to their corresponding records in the USER table using a subquery with IN. Besides, I use SET SQL_SAFE_UPDATES = 0; structure to temporarily disable MySQL's safe update mode, allowing me to run an update that uses a subquery in the `WHERE` clause. This bypasses the restriction only when necessary and ensures the database remains protected by re-enabling safe update mode immediately afterward.
- **SQL Statement:**

```
SET SQL_SAFE_UPDATES = 0;

UPDATE USER
SET IS_ACTIVE = FALSE
WHERE USER_ID IN (
   SELECT USER_ID FROM ENROLLMENT WHERE STATUS = 'Dropped'
);
```

SET SQL_SAFE_UPDATES = 1;

- **Screenshot:**
  SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE	IS_ACTIVE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01	1
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20	1
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15	1
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10	1
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) In this section, you are required to write SQL queries to interact with the database you implemented.

a) Write an SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY.

- List user names and the titles of the courses they are enrolled in. Order the result by the user's join date (newest first).
- **SQL Statement:**

```
SELECT USER.NAME AS UserName, COURSE.TITLE AS CourseTitle, USER.JOIN_DATE
FROM USER
INNER JOIN ENROLLMENT ON USER.USER_ID = ENROLLMENT.USER_ID
INNER JOIN COURSE ON ENROLLMENT.COURSE_ID = COURSE.COURSE_ID
ORDER BY USER.JOIN_DATE DESC;
```

UserName	CourseTitle	JOIN_DATE
Lily Chen	Robotics	2024-02-10
Nguyen Tuan	Machine Learning	2024-01-01
Nguyen Tuan	AI Ethics	2024-01-01
James Walker	AI Ethics	2023-12-20
Hiro Tanaka	Web Development	2023-11-15
Hiro Tanaka	Machine Learning	2023-11-15
Sarah Kim	Philosophy of Mind	2023-10-30

b) Write an SQL query to demonstrate the use of SELECT with WHERE and IN.

- Show all assignments for courses in the categories 'Data Science' or 'Programming'.

- **SQL Statement:**

```
SELECT ASSIGNMENT.TITLE AS AssignmentTitle, COURSE.CATEGORY
FROM ASSIGNMENT
JOIN COURSE ON ASSIGNMENT.COURSE_ID = COURSE.COURSE_ID
WHERE COURSE.CATEGORY IN ('Data Science', 'Programming');
```

AssignmentTitle	CATEGORY
ML Quiz 1	Data Science
Build a Web Page	Programming

c) Write an SQL query to demonstrate the use of at least one DATE function.

- Find users who joined in the last year.
- **SQL Statement:**

```
SELECT NAME, EMAIL, JOIN_DATE
FROM USER
WHERE YEAR(JOIN_DATE) = YEAR(CURDATE()) - 1;
```

NAME	EMAIL	JOIN_DATE
Nguyen Tuan	tuan@gmail.com	2024-01-01
Lily Chen	lily@gmail.com	2024-02-10

d) Write an SQL statement to create a VIEW using a SELECT statement with a JOIN.

- Create a view that shows course titles and their instructors.
- **SQL Statement:**

```
CREATE VIEW COURSE_INSTRUCTORS AS
SELECT COURSE.TITLE AS CourseTitle, INSTRUCTOR.NAME AS InstructorName
FROM COURSE
JOIN INSTRUCTOR ON COURSE.INSTRUCTOR_ID = INSTRUCTOR.INSTRUCTOR_ID;

SELECT * FROM COURSE_INSTRUCTORS;
```

CourseTitle	InstructorName
Machine Learning	Dr. Alice Smith
AI Ethics	Prof. John Lee
Web Development	Dr. Emma Brown
Robotics	Dr. Satoshi Yamamoto
Philosophy of Mind	Prof. Sarah Green

16 Online Quiz 2

Complete Online Quiz 2 available on the unit site

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

Outcome	Weight
Reflection	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

Outcome	Weight
Research and critical review	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

Date	Author	Comment
2025/04/23 13:46	Thi Ngo	Working On It
2025/05/14 19:00	Thi Ngo	Ready to Mark
2025/05/18 13:24	Mohammad Hossain	Good. However, to provide the evidence that you have completed the quiz by yourself, please take a screenshot of the score that include your name. Thanks.
2025/05/18 13:24	Mohammad Hossain	Fix and Resubmit
2025/05/19 13:49	Thi Ngo	Ready to Mark
2025/05/19 13:51	Thi Ngo	Dear Dr. Mohammad,I've taken a new screenshot of my quiz score that clearly shows my name, as you requested. Thank you for your feedback.Best Regards,Suong Ngo.
2025/05/22 12:58	Mohammad Hossain	Excellent work. Thanks.
2025/05/22 12:58	Mohammad Hossain	Complete

Deakin University

Database Fundamentals

OnTrack Submission

Online Quiz 2

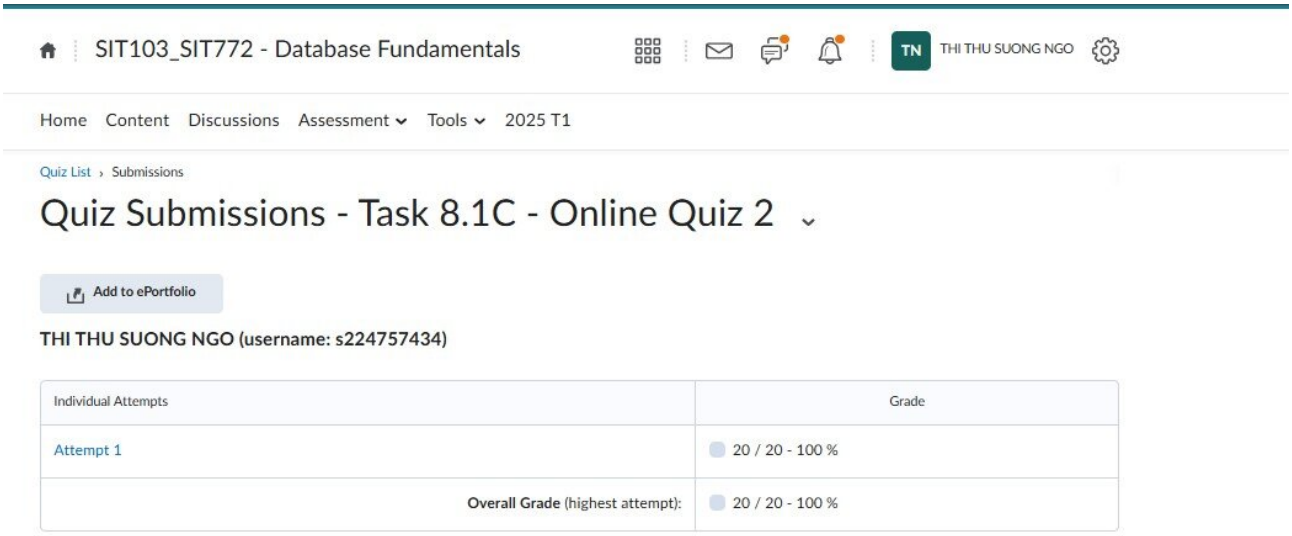
Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/19 13:49

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

In this task, I learned fundamental SQL concepts and commands used in managing relational databases. I explored how to retrieve, insert, update, and delete data using SQL statements, and understood key concepts like JOIN, TRIGGER, FOREIGN KEY, and PL/SQL. I also learned the differences between static and dynamic SQL, how default values and aliases work, and how procedural extensions like PL/SQL enable more complex logic inside the database. Overall, this task helped reinforce my understanding of both basic and advanced database operations and best practices.

May 19, 2025


SIT103_SIT772 - Database Fundamentals
TN
THI THU SUONG NGO
Home
Content
Discussions
Assessment
Tools
2025 T1
Quiz List › Submissions
Quiz Submissions - Task 8.1C - Online Quiz 2
Add to ePortfolio
THI THU SUONG NGO (username: s224757434)
Individual Attempts
Grade
Attempt 1
20 / 20 - 100 %
Overall Grade (highest attempt):
20 / 20 - 100 %

17 PL/PSQL - Trigger, Procedure and Function

Write procedure and Cursors to manipulate SQL results

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

Outcome	Weight
Reflection	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this

by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

Date	Author	Comment
2025/05/06 10:54	Thi Ngo	Working On It
2025/05/14 21:03	Thi Ngo	Ready to Mark
2025/05/18 12:48	Mohammad Hossain	The answer of Q1, 2,3,4, and 5 are not fully correct. Please see the instructions and fix them to resubmit.
2025/05/18 12:48	Mohammad Hossain	Fix and Resubmit
2025/05/19 16:59	Thi Ngo	Ready to Mark
2025/05/19 17:00	Thi Ngo	Dear Dr. Mohammad,I have reviewed the feedback and updated my submission for Task 8.2D accordingly. Specifically, I have revised the answers for Questions 1, 2, 3, 4, and 5 to align with the task instructions, including the correct SQL code and required screenshots before and after execution.Please let me know if anything further needs to be adjusted.Best Regards,Suong Ngo.
2025/05/22 12:12	Mohammad Hossain	Please show the whole table in the output for Q1 to 4.
2025/05/22 12:12	Mohammad Hossain	Otherwise the output are confusing.
2025/05/22 12:12	Mohammad Hossain	Fix and Resubmit
2025/05/22 19:16	Thi Ngo	Need Help
2025/05/22 19:16	Thi Ngo	I am flying back to my home country today so I am currently having limited time to fix the task while it only allows me for 4 hours to fix. Appreciate for your support to allow me submit by the end of tomorrow to ensure the quality outcome of the task. Thank you.
2025/05/23 20:03	Thi Ngo	Working On It
2025/05/23 20:03	Thi Ngo	Ready to Mark
2025/05/23 20:05	Thi Ngo	Dear Dr. Mohammad,I've completed the required updates for Task 8.2D, including revisions to Question 1 to Question 4 as per the feedback. I've ensured that all outputs now include the full CUSTOMER and INVOICE tables before and after each operation, along with the correct SQL code and screenshots.Please let me know if there's anything further needed.Kind Regards,Suong Ngo.
2025/05/25 06:15	Mohammad Hossain	Looks good now.
2025/05/25 06:15	Mohammad Hossain	Complete
2025/05/25 15:52	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

PL/PSQL - Trigger, Procedure and Function

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/23 20:03

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through Tasks 1 to 5, I gained hands-on experience with PL/SQL in Oracle. In Task 1, I learned to create a trigger that automatically updates a customer's balance when a new invoice is inserted. In Task 2, I created a stored procedure to add a new customer using IN parameters. Task 3 extended this by creating a procedure to insert a new invoice and observing how it triggered balance updates. In Task 4, I developed a DELETE trigger that subtracts the invoice amount from a customer's balance when an invoice is removed. Finally, in Task 5, I created a function to return the number of invoices for a specific customer, applying logic with parameters and return values. These tasks helped reinforce my understanding of triggers, procedures, functions, and table manipulation in a database environment.

May 23, 2025

8.2D: PL/SQL (Trigger, Stored Procedure and Function)

Verify the Data from `8_2D_data.sql`

SELECT * FROM CUSTOMER;

```
SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne                1050.11
      1001 Ortega          Juan                   840.92
```

SELECT * FROM INVOICE;

```
   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-22     235.89
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
```

Task 1: Trigger - TRG_UPDATE_CUST_BALANCE

- **Show Full Table before Inserting**

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne                1050.11
      1001 Ortega          Juan                   840.92
```

SELECT * FROM INVOICE;

```
   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-22     235.89
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
```

- **Code to create the trigger**

```
CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE
AFTER INSERT ON INVOICE
FOR EACH ROW
BEGIN
  UPDATE CUSTOMER
  SET CUST_BALANCE = CUST_BALANCE + :NEW.INV_AMOUNT
  WHERE CUST_NUM = :NEW.CUST_NUM;
END;
/
```

```
SQL> CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE
AFTER INSERT ON INVOICE
FOR EACH ROW
BEGIN
  UPDATE CUSTOMER
  SET CUST_BALANCE = CUST_BALANCE + :NEW.INV_AMOUNT
  WHERE CUST_NUM = :NEW.CUST_NUM;
END;
/  2    3    4    5    6    7    8    9

Trigger created.
```

- **Insert the Test Invoice**

```
INSERT INTO INVOICE VALUES (8005, 1001, TO_DATE('27-Apr-2022', 'DD-Mon-YYYY'), 225.40);
```

```
SQL> INSERT INTO INVOICE VALUES (8005, 1001, TO_DATE('27-Apr-2022', 'DD-Mon-YYYY'), 225.40);

1 row created.
```

- **Check full table after inserting**

```
SELECT * FROM INVOICE;
```

```
   INV_NUM   CUST_NUM INV_DATE   INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-22     235.89
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
      8005       1001 27-APR-22      225.4

6 rows selected.
```

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne                1050.11
      1001 Ortega          Juan                  1066.32
```

Task 2: Procedure - PRC_ADD_CUSTOMER

- **Show CUSTOMER Table Before Insert**

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne                1050.11
      1001 Ortega          Juan                  1066.32
```

- **Create the Procedure**

```
CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER (
  p_cust_num     IN NUMBER,
  p_cust_lname   IN VARCHAR2,
  p_cust_fname   IN VARCHAR2,
  p_cust_balance IN NUMBER
)
IS
BEGIN
  INSERT INTO CUSTOMER (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE)
  VALUES (p_cust_num, p_cust_lname, p_cust_fname, p_cust_balance);
END;
/
```

```
SQL> SELECT * FROM CUSTOMER WHERE CUST_NUM = 1002;

no rows selected

SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER (
  2    p_cust_num     IN NUMBER,
  p_cust_lname   IN VARCHAR2,
  p_cust_fname   IN VARCHAR2,
  p_cust_balance IN NUMBER
)
IS
BEGIN
  INSERT INTO CUSTOMER (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE)
  VALUES (p_cust_num, p_cust_lname, p_cust_fname, p_cust_balance);
END;
/
  3    4    5    6    7    8    9   10   11   12
Procedure created.
```

- **Execute the Procedure**

EXEC PRC_ADD_CUSTOMER(1002, 'Rauthor', 'Peter', 0.0);

```
SQL> EXEC PRC_ADD_CUSTOMER(1002, 'Rauthor', 'Peter', 0.0);

PL/SQL procedure successfully completed.
```

- **Show CUSTOMER Table After Insert**

SELECT * FROM CUSTOMER;

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	0

Task 3: Procedure - PRC_ADD_INVOICE

- **Show existing table before executing the procedure to add new invoice**

SELECT * FROM INVOICE;

```
   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-22     235.89
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
      8005       1001 27-APR-22      225.4

6 rows selected.
```

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                      0
```

- **Create the Procedure**

```
CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE (
  p_inv_num     IN NUMBER,
  p_cust_num    IN NUMBER,
  p_inv_date    IN DATE,
  p_inv_amount  IN NUMBER
)
IS
BEGIN
  INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  VALUES (p_inv_num, p_cust_num, p_inv_date, p_inv_amount);
END;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE (
  p_inv_num     IN NUMBER,
  p_cust_num    IN NUMBER,
  p_inv_date    IN DATE,
  p_inv_amount  IN NUMBER
)
IS
BEGIN
  INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  VALUES (p_inv_num, p_cust_num, p_inv_date, p_inv_amount);
END;
/
  2    3    4    5    6    7    8    9   10   11   12
Procedure created.
```

- **Execute the Procedure**

EXEC PRC_ADD_INVOICE(8006, 1002, TO_DATE('29-Apr-2022', 'DD-Mon-YYYY'), 175.85);

```
SQL> EXEC PRC_ADD_INVOICE(8006, 1002, TO_DATE('29-Apr-2022', 'DD-Mon-YYYY'), 175.85);

PL/SQL procedure successfully completed.
```

- **Show all table after the procedure to add new invoice**

SELECT * FROM INVOICE;

```
   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-22     235.89
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
      8005       1001 27-APR-22      225.4
      8006       1002 29-APR-22     175.85

7 rows selected.
```

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                 175.85
```

Task 4: Trigger - TRG_UPDATE_CUST_BALANCE2

- **Show Tables Before Deletion**

SELECT * FROM CUSTOMER;

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	175.85

SELECT * FROM INVOICE;

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-22	235.89
8001	1001	23-MAR-22	312.82
8002	1001	30-MAR-22	526.1
8003	1000	12-APR-22	194.78
8004	1000	23-APR-22	619.44
8005	1001	27-APR-22	225.4
8006	1002	29-APR-22	175.85

7 rows selected.

- **Create the DELETE Trigger**

```
CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE2

AFTER DELETE ON INVOICE

FOR EACH ROW

BEGIN

  UPDATE CUSTOMER

  SET CUST_BALANCE = CUST_BALANCE - :OLD.INV_AMOUNT

  WHERE CUST_NUM = :OLD.CUST_NUM;

END;
```

/

```
SQL> CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE2
AFTER DELETE ON INVOICE
FOR EACH ROW
BEGIN
  UPDATE CUSTOMER
  SET CUST_BALANCE = CUST_BALANCE - :OLD.INV_AMOUNT
  WHERE CUST_NUM = :OLD.CUST_NUM;
END;
/  2    3    4    5    6    7    8    9

Trigger created.
```

- **Delete the Invoice**

DELETE FROM INVOICE WHERE INV_NUM = 8000;

```
SQL> DELETE FROM INVOICE WHERE INV_NUM = 8000;

1 row deleted.
```

- **Show Tables After Deletion**

SELECT * FROM INVOICE;

```
   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8001       1001 23-MAR-22     312.82
      8002       1001 30-MAR-22      526.1
      8003       1000 12-APR-22     194.78
      8004       1000 23-APR-22     619.44
      8005       1001 27-APR-22      225.4
      8006       1002 29-APR-22     175.85

6 rows selected.
```

SELECT * FROM CUSTOMER;

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne                814.22
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                 175.85
```

Task 5: Function - GET_INVOICE_COUNT

- **Create the Function**

```
CREATE OR REPLACE FUNCTION GET_INVOICE_COUNT (
  p_cust_num IN NUMBER
) RETURN NUMBER
IS
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count
  FROM INVOICE
  WHERE CUST_NUM = p_cust_num;

  RETURN v_count;
END;
/
```

```
SQL> CREATE OR REPLACE FUNCTION GET_INVOICE_COUNT (
  p_cust_num IN NUMBER
) RETURN NUMBER
IS
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count
  FROM INVOICE
  WHERE CUST_NUM = p_cust_num;

  RETURN v_count;
END;
/  2    3    4    5    6    7    8    9   10   11   12   13

Function created.
```

- **Test the Function**

```
SELECT GET_INVOICE_COUNT(1000) FROM DUAL;
```

```
GET_INVOICE_COUNT(1000)
-----------------------
                      2
```

SELECT GET_INVOICE_COUNT(1001) FROM DUAL;

```
SQL> SELECT GET_INVOICE_COUNT(1001) FROM DUAL;

GET_INVOICE_COUNT(1001)
-----------------------
                      3
```

SELECT GET_INVOICE_COUNT(1002) FROM DUAL;

```
SQL> SELECT GET_INVOICE_COUNT(1002) FROM DUAL;

GET_INVOICE_COUNT(1002)
-----------------------
                      1
```

18 Data Analysis and Visualization using Excel

Use Excel PIVOT TABLE/CHART feature to do some basic Business Intelligence

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence (BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence (BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence (BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

Outcome	Weight
Reflection	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence

(BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

Outcome	**Weight**
Research and critical review	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence (BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

Date	**Author**	**Comment**
2025/05/06 10:54	Thi Ngo	Working On It
2025/05/15 15:46	Thi Ngo	Ready to Mark
2025/05/18 12:16	Mohammad Hossain	Well-done.
2025/05/18 12:16	Mohammad Hossain	Complete
2025/05/19 17:46	Thi Ngo	Thank you very much for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Data Analysis and Visualization using Excel

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/15 15:46

Tutor:
Mohammad belayet Hossain

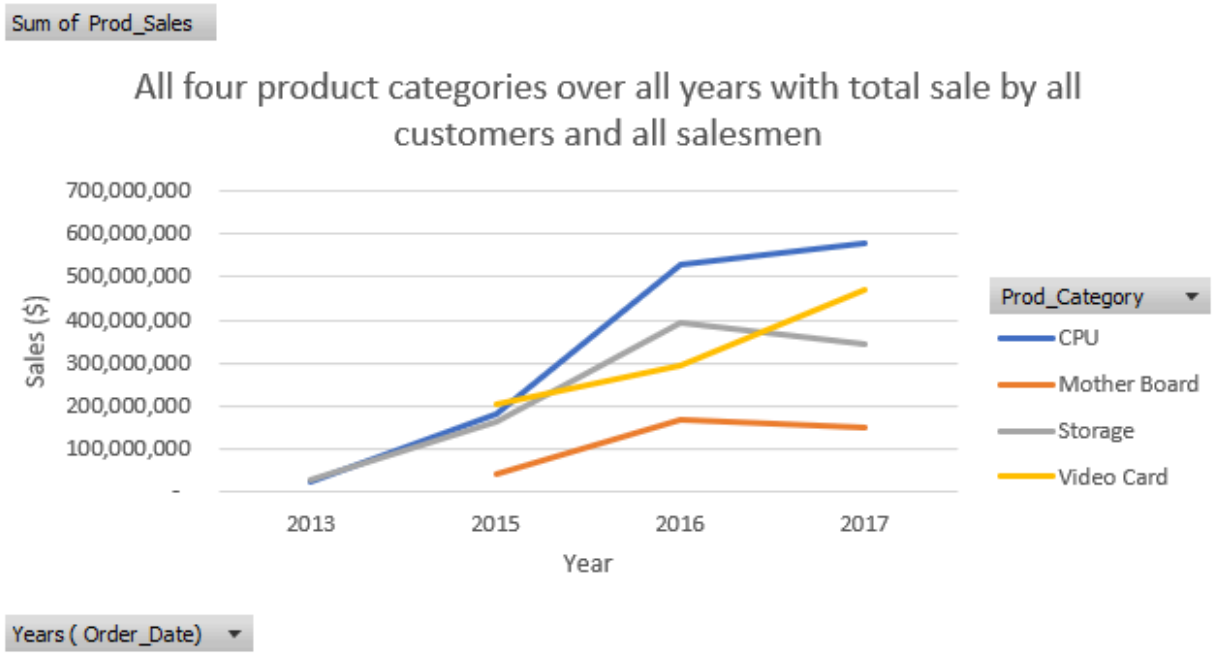
Outcome	**Weight**
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

This task enhanced my understanding of data analysis and its application in real-world business and educational contexts. Through the Excel component, I learned to effectively use PivotTables and PivotCharts to summarize and visualize complex sales data. I developed skills in filtering data, identifying trends across years and product categories, and creating interactive dashboards that support informed decision-making.Additionally, by researching and writing the Coursera Business Intelligence (BI) report, I deepened my knowledge of how digital platforms leverage data for strategic growth. I explored how Coursera collects user data, applies BI tools to personalize learning, supports instructors through performance dashboards, and delivers enterprise-level reporting. I also gained insight into real-time analytics for platform optimization and global marketing strategies.Overall, the task strengthened my technical, analytical, and academic communication skills while illustrating the value of BI in both operational efficiency and educational impact.

May 15, 2025

Task 1: Line Chart - Product Sales Trends Over Time

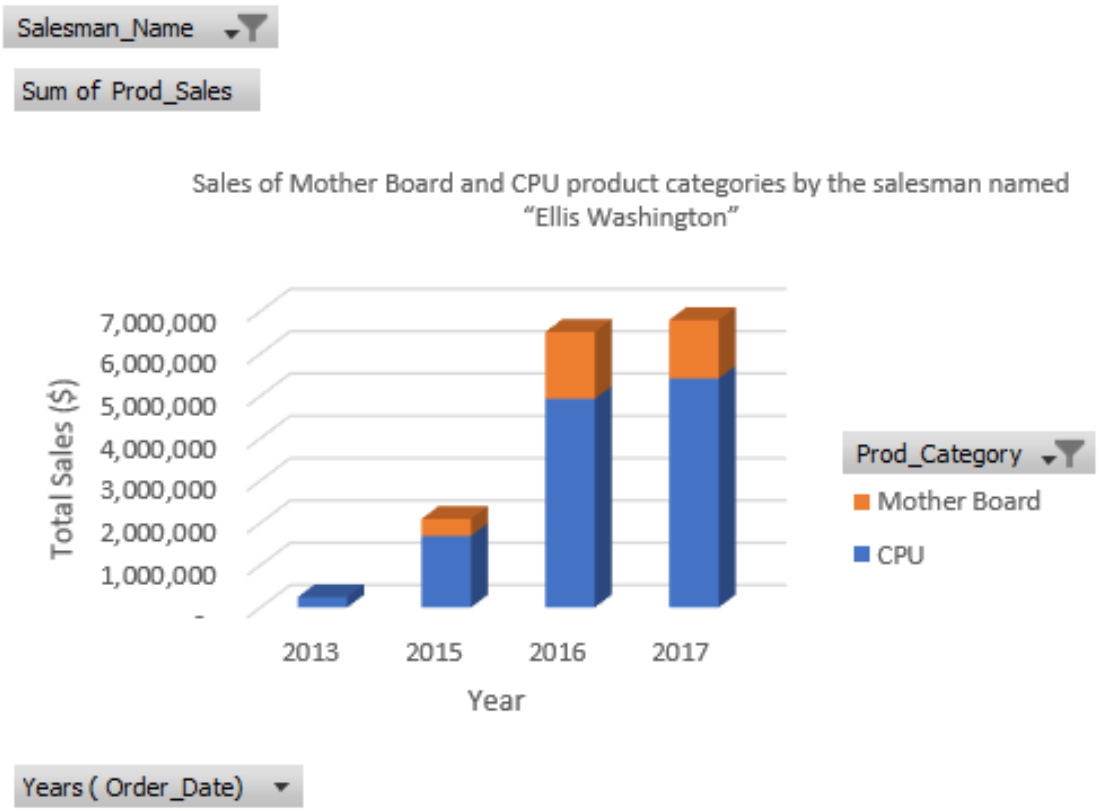
Sum of Prod_Sales	Prod_Category				
Years (Order_Date	CPU	Mother Board	Storage	Video Card	Grand Total
2013	26,048,909		28,977,757		55,026,666
2015	180,680,057	43,133,972	165,699,395	205,807,589	595,321,012
2016	527,668,509	169,049,508	392,957,361	292,935,609	1,382,610,986
2017	578,710,188	148,076,485	343,175,709	471,713,807	1,541,676,189
Grand Total	**1,313,107,663**	**360,259,964**	**930,810,223**	**970,457,004**	**3,574,634,854**



This line chart visualizes the total sales of all four product categories (CPU, Mother Board, Storage, and Video Card) from 2013 to 2017. The chart includes sales data from all customers and all salesmen to show the overall trend for each category over time.

Task 2: 3-D Stacked Column Chart – Sales by Ellis Washington

Salesman_Name	Ellis Washingtor		
Sum of Prod_Sales	**Prod_Category**		
Years (Order_Date	**CPU**	**Mother Board**	**Grand Total**
2013	243,448		243,448
2015	1,688,599	403,121	2,091,720
2016	4,931,481	1,579,902	6,511,383
2017	5,408,506	1,383,892	6,792,399
Grand Total	**12,272,034**	**3,366,916**	**15,638,950**



This 3-D stacked column chart shows the yearly sales of CPU and Mother Board products made by the salesman Ellis Washington from 2013 to 2017. Other product categories and salesmen are excluded to isolate the performance of one individual and two product types.

Task 3: Coursera Business Intelligence Report

1. Introduction

In the rapidly evolving field of online education, data-driven decision-making has become essential for maintaining competitive advantage. This report focuses on Coursera, a global online learning platform, and examines how it leverages data collection and Business Intelligence (BI) tools to support personalization, instructional improvement, and strategic growth. By analyzing Coursera's use of BI, the report highlights the role of data in shaping learner experience and driving the platform's global success.

2. Company Overview

Coursera is a global online learning platform founded in 2012 by Stanford professors Andrew Ng and Daphne Koller. Since its inception, Coursera has grown into one of the most recognized providers of Massive Open Online Courses (MOOCs), collaborating with over 300 leading universities and companies worldwide. The platform aims to make high-quality education accessible to anyone, anywhere. Its offerings range from short courses and professional certificates to advanced specializations and full online degree programs, catering to individual learners, institutions, and enterprises alike (Coursera, 2024).

3. Core Business Model

Coursera operates on a freemium and subscription-based model, allowing users to access course content for free while charging for graded assessments and verified certificates. Revenue is generated through one-time payments for courses and specializations, as well as Coursera Plus- a subscription offering unlimited access to most content (Coursera, 2024). The platform also offers fully accredited online degree programs in partnership with universities, providing cost-effective alternatives to traditional education (Shah, 2021). Beyond individual learners, Coursera serves enterprise clients through Coursera for Business, Government, and Campus. These services provide tailored learning paths and analytics dashboards for workforce development and institutional training (Kemp & Grieve, 2014).

4. Data Collection

To support continuous improvement and personalization, Coursera collects a broad spectrum of user data. This includes demographic information such as age, location, and gender (when disclosed), along with behavioral data like video engagement, course progress, and clickstream activity (Huang et al., 2020). Assessment data- such as quiz scores and assignment results- are used to evaluate learning outcomes and inform course refinement (Veletsianos & Shepherdson, 2016). Coursera also gathers user feedback through surveys and ratings, providing insights into learner satisfaction (Coursera, 2023). In addition, technical data, including device type and session logs, aid in platform optimization and performance monitoring (Ifenthaler & Yau, 2020). Collectively, this data ecosystem enables Coursera to enhance content quality, personalize learning experiences, and maintain system reliability- laying the foundation for effective use of Business Intelligence tools.

4. Use of Business Intelligence

Business Intelligence (BI) is a foundational element in Coursera's operational and strategic framework. By integrating BI tools with advanced analytics and machine learning, Coursera continuously refines its educational offerings, optimizes user engagement, and informs business decisions across all levels of the organization.

4.1) Personalized Course Recommendations

Coursera applies Business Intelligence (BI) and artificial intelligence (AI) to provide personalized course recommendations based on learners' individual preferences, prior activity, and performance history. Behavioral data- including quiz scores, course progress, and browsing behavior- is collected and analyzed using machine learning algorithms to generate tailored suggestions (Zhou et al., 2020). These recommendation systems support adaptive learning pathways that enhance learner engagement, reduce dropout rates, and improve overall satisfaction. Studies have shown that personalization significantly increases the likelihood of course completion and conversion to paid services, such as certificates and subscriptions (Nguyen, 2022; Sinha et al., 2014). The below diagram showcases the flow of data within a course recommendation system, highlighting components such as user input, interface, model fitting, and system output.

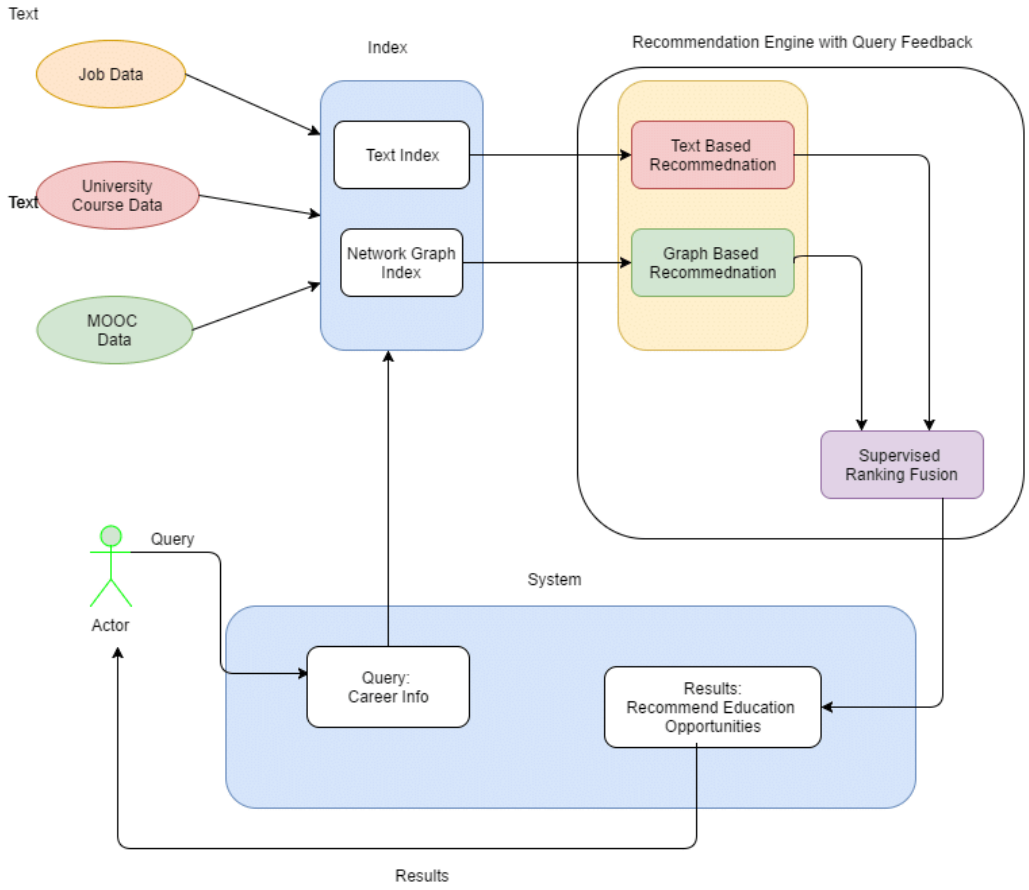


Figure 1: The impact of visual and interactive course recommendation in university environments (Ma et al., 2021).

4.2) Instructor and Course Performance Feedback

Coursera leverages Business Intelligence (BI) dashboards to provide instructors and course teams with real-time insights into learner behavior and course effectiveness. These dashboards aggregate data on learner engagement, assessment scores, dropout patterns, and time spent per learning unit. Such analytics enable instructors to identify underperforming modules, address content gaps, and adapt pedagogical strategies accordingly (Ifenthaler & Yau, 2020). Additionally, metrics such as course completion rates, learner ratings, and qualitative feedback are used by Coursera to inform decisions on course updates, retirements, or promotional placement (Coursera, 2024). These BI-driven feedback systems support continuous improvement and help maintain content relevance and instructional quality at scale (Veletsianos & Shepherdson, 2016).

4.3) B2B and Institutional Reports

Coursera provides Business Intelligence (BI) tools to enterprise and institutional clients through platforms like Coursera for Business and Coursera for Campus. These dashboards offer real-time insights into learner engagement, course completion rates, and skill development, enabling organizations to assess training outcomes and identify workforce or academic gaps (Coursera, 2024). BI reports support data-driven decisions by aligning learning initiatives with strategic goals and improving program effectiveness (Ifenthaler & Yau, 2020). Research indicates that such analytics enhance organizational learning efficiency and adaptability at scale (Chatti et al., 2012).

4.4) Global Marketing Optimization

Business Intelligence (BI) is central to Coursera's global marketing strategy, enabling data-driven decisions across digital campaigns. By analyzing key metrics such as website traffic sources, click-through rates, and conversion data, Coursera identifies high-performing marketing channels and user preferences across regions (Statista, 2023). These insights support content localization, targeted outreach, and pricing adjustments tailored to specific markets. Such optimization enhances return on investment (ROI) and broadens global reach. Research supports the effectiveness of BI in digital marketing, highlighting its role in improving customer segmentation, campaign efficiency, and market responsiveness (Chaffey & Ellis-Chadwick, 2019; Wedel & Kannan, 2016).

4.5) Strategic Growth and Real-Time Decision-Making

Coursera utilizes real-time Business Intelligence (BI) dashboards to monitor platform performance, user behavior, and content engagement trends. These systems enable technical teams to detect and resolve operational issues promptly, ensuring platform stability and responsiveness, particularly during periods of high demand (Ifenthaler & Yau, 2020). Strategically, BI supports Coursera in identifying growth opportunities by analyzing regional demand, emerging skills, and learner preferences. This allows the platform to expand its course offerings, establish new institutional partnerships, and allocate resources to high-impact areas (Chatti et al., 2012). Research affirms that real-time analytics enhance agility in digital platforms, supporting proactive decision-making and competitive positioning in dynamic markets (Wang et al., 2018).

5. Conclusion

Coursera's effective integration of data collection and Business Intelligence (BI) serves as a foundational driver of its success in the online education industry. By leveraging BI tools, the platform delivers personalized learning pathways, supports continuous course enhancement, and informs targeted marketing strategies. Furthermore, its enterprise analytics empower institutions and organizations to track learning outcomes and align training with strategic objectives. This data-driven approach enables Coursera to scale its operations while maintaining relevance and impact. Ultimately, the strategic use of BI reinforces Coursera's position as a leading provider of accessible, adaptive, and globally scalable digital education.

References

Alhijawi, B. and Kilani, Y. (2018). 'Course Recommender System Architecture', *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(9), pp. 410–416. Available at: https://www.researchgate.net/figure/Course-Recommender-System-Architecture_fig1_326580635 [Accessed 15 May 2025].

Coursera (2023). Privacy Notice. Available at: https://www.coursera.org/about/privacy [Accessed 15 May 2025].

Coursera (2024). *About Coursera*. Available at: https://www.coursera.org/about [Accessed 15 May 2025].

Coursera (2024). *Enterprise Solutions*. Available at: https://www.coursera.org/business [Accessed 15 May 2025].

Chatti, M.A., Dyckhoff, A.L., Schroeder, U. and Thüs, H. (2012). 'A reference model for learning analytics', *International Journal of Technology Enhanced Learning*, 4(5/6), pp.318–331.

Chaffey, D. and Ellis-Chadwick, F. (2019). *Digital Marketing*. 7th ed. Harlow: Pearson Education.

Nguyen, A. (2022). 'How Coursera uses AI for personalizing learning experiences', *Journal of Online Education Technology*, 14(3), pp.45–56.

Ma, B., Lu, M., Taniguchi, Y. and Konomi, S. (2021). 'CourseQ: The impact of visual and interactive course recommendation in university environments', *Sensors*, 21(13), p.4393. doi:10.3390/s21134393.

Huang, G., Hew, K.F. and Fryer, L.K. (2020). 'Learner engagement in MOOCs: Scale development and validation', The Internet and Higher Education, 45, pp.1–12. doi:10.1016/j.iheduc.2019.100728.

Statista ( ). *Online Learning Platform User Insights*. Available at: https://www.statista.com/ [Accessed 15 May 2025].

Shah, D. (2021). *Class Central*. Available at: https://www.classcentral.com/report/mooc-stats-2021/ [Accessed 15 May 2025].

Sinha, T., Li, N. and Jermann, P. (2014). 'Capturing "attrition intensifying" structural traits from didactic interaction sequences of MOOC learners', *arXiv preprint arXiv:1409.5887*.

Wang, Y., Kung, L. and Byrd, T.A. (2018). 'Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations', *Technological Forecasting and Social Change*, 126, pp.3–13.

Wedel, M. and Kannan, P.K. (2016). 'Marketing analytics for data-rich environments', *Journal of Marketing*, 80(6), pp.97–121.

Zhou, Y., Ji, S. and Ma, Y. (2020). 'Deep learning-based recommender system: A survey and new perspectives', *ACM Computing Surveys*, 53(1), pp.1–38.

Ifenthaler, D. and Yau, J.Y-K. (2020). 'Utilising learning analytics for study success: Reflections on current empirical findings', *International Journal of Learning Analytics and Artificial Intelligence for Education*, 2(1), pp.29–45.

Kemp, N. and Grieve, R. (2014). *Frontiers in Psychology*, 5, 1278.

Veletsianos, G. and Shepherdson, P. (2016). 'A systematic analysis and synthesis of the empirical MOOC literature published in 2013–2015', *The International Review of Research in Open and Distributed Learning*, 17(2), pp.198–221

19 Database security

In this task, you will learn to analyze a database security scenario to identify operational vulnerabilities

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

Outcome	Weight
Reflection	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

Date	**Author**	**Comment**
2025/05/15 15:48	Thi Ngo	Working On It
2025/05/20 22:15	Thi Ngo	Ready to Mark
2025/05/21 19:09	Mohammad Hossain	Very good.
2025/05/21 19:09	Mohammad Hossain	Complete

Deakin University

Database Fundamentals

OnTrack Submission

Database security

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/20 22:15

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through this task, I deepened my understanding of how ostensibly minor operational oversights-such as unsecured physical access, unmanaged removable media, weak authentication protocols, and unpatched systems can cumulatively undermine an organisation's security posture. I learned the critical importance of implementing layered technical controls (e.g., full-disk encryption, multi-factor authentication, and automated patch management) alongside rigorous procedural measures (such as asset tracking, incident reporting, and ongoing staff training). Finally, analysing the Medibank breach reinforced the value of continuous monitoring, least-privilege access, and independent security audits in detecting and mitigating sophisticated attacks before they escalate into large-scale data compromises.

May 20, 2025

Report: Database Security Assessment for Sunrise Financial Solutions and Medibank Real Case Data Breach Incident

Executive Summary

This report critically examines the operational security deficiencies present at Sunrise Financial Solutions, a mid-sized accounting firm, and delineates a suite of targeted countermeasures. Section 1 systematically delineates five principal vulnerabilities derived from the case materials. Section 2 articulates three technical and procedural safeguards designed to preserve data confidentiality in the event of device loss or misappropriation. Section 3 prescribes a triad of email-authentication and anti-phishing controls to enhance the veracity of internal communications. Section 4 evaluates three secure alternatives to the use of personal USB media for file exchange. Finally, Section 5 presents an in-depth case study of the October 2022 Medibank breach, detailing the incident timeline, underlying attack vectors, resultant impacts, and salient lessons for the prevention of analogous security failures.

A- Sunrise Financial Solutions Case Study

1. Identification of Security Vulnerabilities

- **Unsecured Physical Access**

Sunrise Financial Solutions' operational environment permits unrestrained physical entry, as employees routinely leave office doors for convenience . This practice undermines perimeter integrity and allows unauthorized individuals to access sensitive workstations and documents, thereby elevating the risk of data theft, device sabotage, or other malicious activities.

  - *Evidence:* "Employees frequently… leaving their office doors open for easy access" .
  - *Risk:* Unauthorized individuals may enter the premises, leading to data theft or device tampering.

- **Use of Personal USB Drives**

The firm's data-transfer processes include the regular use of personal USB flash drives-most notably by a senior accountant who archives client financial records on an unsecured

personal device, and by interns who exchange files via the same medium . Such unmanaged devices are prone to malware infection and loss, which can precipitate the unauthorized dissemination or corruption of confidential information.

- *Evidence:* “Senior accountant… backs up client financial records on his personal flash drive” and “interns… regularly exchange files through USB drives” corbado.com.
- *Risk:* Personal drives may be infected with malware or lost, exposing sensitive data.

- **Weak Password Policy**

Current authentication protocols permit- and even tacitly encourage- the creation of trivial passwords based on easily obtainable personal information; most staff use their birth year, and the IT support manager has recommended short passwords for simplicity

- *Evidence:* “Most employees use their birth year in passwords” and “IT Support Manager… advised staff to set short passwords” upguard.com.
- *Risk:* Low-entropy passwords are trivial to brute-force or guess, granting attackers easy access.

- **Unpatched Systems**

Critical workstations within the organization have operated for extended periods without installation of recent security updates, with no remedial measures undertaken despite identified vulnerabilities . Failure to apply timely patches leaves known software flaws unaddressed, providing adversaries with exploitable vectors for unauthorized intrusion and privilege escalation.

- *Evidence:* “One of the office’s main computers… has been operating without recent security updates, but… no immediate action has been taken” Reuters.
- *Risk:* Known vulnerabilities remain exploitable, increasing the likelihood of compromise.

- **Lack of VPN for Remote Access**

Executive personnel routinely connect to corporate resources without the protection of a Virtual Private Network (VPN), citing concerns over perceived performance degradation . The absence of encryption on these channels permits interception of login credentials and sensitive data by network eavesdroppers, severely undermining confidentiality and integrity.

- *Evidence:* Directors “prefer not to use a Virtual Private Network (VPN)… as they believe it slows down their efficiency” .
- *Risk:* Remote connections over unencrypted channels can be intercepted, exposing credentials and data.

2. Safeguards Against Device Loss or Theft

- **Full-Disk Encryption**

All portable endpoints- including laptops and removable storage- should employ disk-level encryption such as Microsoft BitLocker or Apple FileVault. By encrypting the entire volume, data at rest remains indecipherable without the appropriate cryptographic keys, thereby preventing unauthorized access in the event of device loss or theft (Microsoft, 2023; Apple, 2023).

- **Remote Wipe Capability**

An enterprise-grade Mobile Device Management (MDM) platform must be implemented to enforce remote wipe capabilities on both notebook computers and USB-style media. This control enables rapid, centralized eradication of sensitive information from compromised devices, effectively narrowing the window of exposure and limiting potential data exfiltration (Smith & Brooks, 2016; Gartner, 2022).

- **Asset Tracking and Mandatory Reporting**

Corporate hardware should be uniquely tagged- via RFID or serialized asset labels- and recorded within a centralized inventory system. Employees must be mandated to report any instance of device loss or theft within one business hour. Such rigorous tracking and reporting protocols substantially improve recovery prospects and facilitate an organized, timely response to potential security incidents (Nguyen & Patel, 2018; Zhao et al., 2019).

3. Email Authentication and Anti-Phishing Controls

- **Email Authentication via SPF, DKIM, and DMARC**

Enforcing Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication Reporting and Conformance (DMARC) prevents unauthorized mail servers from dispatching messages on behalf of the company domain. SPF specifies which IP addresses are permitted to send mail for a domain, DKIM employs cryptographic signatures to validate message origin, and DMARC provides policy enforcement and reporting mechanisms to reject unauthenticated mail (NIST, 2007; NIST, 2017).

- **Multi-Channel Verification for Credential Requests**

Any internal request for credential validation- such as password resets or access-approval e-mails- must be confirmed through an independent communication channel (e.g., a telephone call or face-to-face inquiry). This out-of-band verification mitigates the risk that a single compromised channel could be used to harvest user credentials (NIST, 2017).

- **Ongoing Security Awareness and Phishing Simulations Training**

Implement a formal training programme that conducts simulated phishing campaigns and refresher sessions at least quarterly. Regular exposure to realistic phishing scenarios reinforces employees' ability to detect malicious emails and fosters enduring adherence to secure handling practices (NIST, 2003; NIST, 2024).

4. Secure Alternatives to Personal USB Drives

- **Centralized, Access-Controlled File Shares**

Deploy network-attached storage (NAS) systems configured with role-based access controls (RBAC) to ensure that users can access only the resources necessary for their responsibilities (NIST, 2020). Centralizing file storage under NAS facilitates comprehensive logging of all file operations and enables integration with enterprise antivirus solutions, thereby ensuring that every file transfer is automatically scanned for malware (ISO/IEC, 2013).

- **Encrypted Cloud Storage**

Utilize enterprise cloud storage services- such as Microsoft Office 365 OneDrive or Amazon S3- that support server-side encryption (SSE) with audit-capable access logging. SSE encrypts data at rest using robust cryptographic algorithms (e.g. AES-256), while immutable audit trails record every access event, aiding in compliance and forensic investigations (AWS, 2023; Microsoft, 2022).

- **Managed Removable Media**

Issue organization-owned USB drives equipped with hardware-based encryption modules and tamper-evident seals in accordance with media protection standards (NIST, 2008). These managed devices enforce encryption at the firmware level and generate usage logs, preserving the portability of removable media without compromising on security governance (ISO/IEC, 2013).

B- Case Study: The Medibank Data Breach (October 2022)

1) Incident Summary

In October 2022, Medibank Private- Australia's largest private health insurer- suffered a catastrophic breach in which threat actors exfiltrated the personal and medical records of approximately 9.7 million current and former customers and, after rejecting a ransom demand, gradually published the stolen data on dark-web forums (Sadler, 2024; Privacy108, 2023). No ransom payment was made; instead, the attackers leveraged their illicit access to inflict maximal harm by publicly releasing sensitive medical histories, including treatment for HIV, abortions and mental-health conditions (Privacy108, 2023).

2) Attack Vector and Exploitation

Subsequent court filings revealed that the breach originated from the compromise of a privileged contractor's credentials, which had been inadvertently synchronized from a work laptop's browser to a personal device (Sadler, 2024). In August 2022, a contractor storing Medibank credentials in a browser profile unwittingly allowed those credentials to propagate to a non-secure personal machine. Attackers then deployed malware to harvest the credentials and verify access to Medibank's Microsoft Exchange server (Sadler, 2024). Two weeks later, using only a username and password- since the organisation had failed to enforce multi-factor authentication (MFA) on its GlobalProtect VPN- adversaries gained direct remote access to the

corporate network (Sadler, 2024). Over a period of days, bespoke scripts automated the extraction of approximately 520 GB of data from Medibank's primary customer database via established backdoors (Sadler, 2024).

3) Impact and Consequences

The ramifications of this breach were manifold. Affected customers faced heightened risks of identity theft, extortion, and psychological distress, owing to the disclosure of deeply personal medical information (ABC News, 2024). Medibank itself incurred direct remediation and indemnity costs totalling A$86.2 million in its 2024 financial year, with the total financial outlay on cyber-incident response- including system hardening, legal fees, customer support and regulatory remediation- projected to exceed A$126 million by mid-2025 (Tech Business News, 2024). Regulatory repercussions also ensued: the Office of the Australian Information Commissioner (OAIC) initiated civil-penalty proceedings, alleging "serious interference" with the privacy of nearly 9.7 million individuals and seeking fines of up to A$2.22 million per contravention under the Privacy Act 1988 (OAIC, 2023).

4) Preventive Measures and Lessons Learned

The Medibank incident underscores the imperative of a layered security architecture grounded in industry best practices:

- **Multi-Factor Authentication (MFA)**

Enforce MFA for all remote and privileged-access channels, particularly VPN gateways and email systems. MFA provides an essential barrier against credential-based attacks, ensuring that possession of a password alone is insufficient for network entry (NIST, 2017).

- **Least-Privilege Access Control**

Implement the principle of least privilege, whereby users receive only the minimal access rights necessary for their roles. Regular privilege reviews and automated access-reconciliation processes prevent excessive entitlements that attackers can exploit once inside the network (NIST, 2020).

- **Robust Patch Management**

Deploy an automated patch-management framework in line with NIST SP 800-40 guidelines. By systematically scanning for known vulnerabilities and applying security updates within a defined service-level agreement (SLA), organisations can curtail the temporal window in which adversaries exploit unpatched software (NIST, 2002).

- **Continuous Monitoring and Incident Response**

Establish a Security Information and Event Management (SIEM) system as recommended by NIST SP 800-137, coupled with a formally documented incident-response plan. Real-time log aggregation and alerting facilitate rapid detection of anomalous behaviour- such as large‑scale data transfers- while predefined response playbooks ensure coordinated containment and remediation (NIST, 2011).

- **Regular Independent Security Audits**

Commission periodic third-party assessments under ISO/IEC 27001:2013 to validate the efficacy of security controls and to uncover latent vulnerabilities. Independent audits provide an objective measure of compliance and drive continual improvement of the information-security management system (ISO/IEC, 2013).

5) Conclusion

The Medibank breach illustrates how fundamental lapses- such as the absence of MFA, reliance on single-factor VPN authentication, and failure to manage patched systems- can culminate in widespread data compromise and profound organisational and societal harm. By adopting layered defences, enforcing principle-based access controls, and institutionalising a culture of proactive security monitoring and independent validation, enterprises can significantly strengthen their resilience against similarly sophisticated threat campaigns.

References

Apple, 2023. *FileVault User Guide*. Apple Inc.

AWS (2023) *Data Protection in Amazon S3*. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html (Accessed: 20 May 2025).

Nguyen, T. & Patel, R., 2018. ‘Enhancing IT Asset Management with RFID Technology’, *Journal of Information Systems*, 32(4), pp. 45–58.

National Institute of Standards and Technology (NIST) (2017) *Trustworthy Email*. Gaithersburg, MD: NIST. NIST

National Institute of Standards and Technology (NIST) (2003) *SP 800-50: Building an Information Technology Security Awareness and Training Program*. Gaithersburg, MD: NIST. NIST Computer Security Resource Center

National Institute of Standards and Technology (NIST) (2024) *SP 800-50r1: Building a Cybersecurity and Privacy Learning Program*. Gaithersburg, MD: NIST.

NIST (2008) *SP 800-111: Guide to Storage Encryption Technologies for End User Devices*. Gaithersburg, MD: National Institute of Standards and Technology.

NIST (2020) *SP 800-53 Revision 5: Security and Privacy Controls for Information Systems and Organizations*. Gaithersburg, MD: National Institute of Standards and Technology.

ABC News (2024) ‘Medibank’s failure exposed nine million customers to identity theft and emotional distress’, *ABC News*, 22 June. Available at: https://www.abc.net.au/news/2024-06-22/medibank-alerts-australia-cybersecurity-breach/104003576 (Accessed: 20 May 2025).

NIST (2002) *SP 800-40 Revision 3: Guide to Enterprise Patch Management Technologies*, Gaithersburg, MD: National Institute of Standards and Technology.

NIST (2011) *SP 800-137: Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*, Gaithersburg, MD: National Institute of Standards and Technology.

NIST (2017) *SP 800-63B: Digital Identity Guidelines – Authentication and Lifecycle Management*, Gaithersburg, MD: National Institute of Standards and Technology.

NIST (2020) *SP 800-53 Revision 5: Security and Privacy Controls for Information Systems and Organizations*, Gaithersburg, MD: National Institute of Standards and Technology.

Microsoft, 2023. *BitLocker Drive Encryption Overview*. Microsoft Corporation.

Microsoft (2022) ‘Overview of encryption in OneDrive and SharePoint’, *Microsoft Docs*. Available at: https://docs.microsoft.com/en-us/office365/securitycompliance/office-365-encryption (Accessed: 20 May 2025).

Gartner, 2022. *Market Guide for Mobile Device Management*. Gartner Research.

Smith, A. & Brooks, D., 2016. ‘Mobile Device Management in Corporate Environments’, *International Journal of Mobile Computing*, 11(2), pp. 101–115.

Sadler, D. (2024) ‘Court docs reveal shocking cause of Medibank breach’, *Information Age*, 18 June. Available at: https://ia.acs.org.au/article/2024/court-docs-reveal-shocking-cause-of-medibank-breach.html (Accessed: 20 May 2025).

Zhao, L., Chen, Y. & Kumar, S., 2019. ‘Optimizing Incident Reporting Procedures for IT Security’, *Journal of Cybersecurity Practices*, 5(1), pp. 12–27.

National Institute of Standards and Technology (NIST) (2007) *SP 800-45 Revision 2: Guidelines on Electronic Mail Security*. Gaithersburg, MD: NIST. NIST Computer Security Resource Center

ISO/IEC (2013) *ISO/IEC 27002:2013 Information Technology- Security Techniques- Code of Practice for Information Security Controls*. Geneva: International Organization for Standardization.

OAIC (2023) ‘OAIC takes civil penalty action against Medibank’, *Office of the Australian Information Commissioner*. Available at: https://www.oaic.gov.au/news/media-centre/oaic-takes-civil-penalty-action-against-medibank (Accessed: 20 May 2025).

Privacy108 (2023) ‘Understanding the OAIC civil penalty against Medibank’, *Privacy108*, 10 December. Available at: https://privacy108.com.au/insights/the-oaic-action-against-medibank/ (Accessed: 20 May 2025).

Tech Business News (2024) ‘Medibank’s data breach costs expected to hit $126 million by mid-2025’, *Tech Business News*, 23 August. Available at: https://www.techbusinessnews.com.au/news/medibanks-data-breach-costs-expected-to-hit-126-million-by-mid-2025/ (Accessed: 20 May 2025).

ISO/IEC (2013) *ISO/IEC 27001:2013 Information Technology- Security Techniques- Information Security Management Systems- Requirements*. Geneva: International Organization for Standardization.

20 Learning Summary Report

Prepare a learning summary report reflect on your learning in the unit against the ULOs

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

Outcome	Weight
Reflection	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a

summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

Date	Author	Comment
2025/05/15 15:49	Thi Ngo	Working On It
2025/05/22 11:28	Thi Ngo	Ready to Mark
2025/05/22 14:10	Mohammad Hossain	Looks good.
2025/05/22 14:11	Mohammad Hossain	Complete
2025/05/26 19:42	Thi Ngo	thank you Dr. Mohammad for your feedback, I highly appreciate it!

Deakin University

Database Fundamentals

OnTrack Submission

Learning Summary Report

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/22 11:28

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Developing this learning reflection portfolio provided me with an opportunity to step back and critically evaluate my progress throughout the unit. It helped me recognise not only the technical skills I've developed- such as data modelling, SQL implementation, and system integration- but also how my thinking has matured around topics like ethical data use, scalability, and professional responsibility.By reflecting on both challenges and achievements, I gained deeper insight into my learning process, identified areas for further growth, and strengthened my ability to communicate complex ideas clearly. This reflection reinforced my confidence in applying database knowledge to real-world contexts and clarified the direction I want to take in my IT career. Ultimately, this portfolio served as both a summary of what I've accomplished and a springboard for how I plan to keep learning and evolving as a future IT professional.

May 22, 2025

SIT772

Database Fundamentals

THI THU SUONG NGO
STUDENT ID: 224757434

Self-Assessment Details
The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment				✔

Self-Assessment Statement

Based on my comprehensive and consistent demonstration of mastery across all Unit Learning Outcomes, I am applying for a High Distinction in SIT772 - Database Fundamentals, Throughout the trimester, I engaged deeply with the course content, extending my learning beyond the core curriculum through critical reflection, independent research, and application of best practices in database design and management.

This portfolio reflects not only successful completion of the required tasks but also evidence of higher-order thinking, integration of contemporary technologies, and strong ethical awareness in handling data. My ability to link theory to practice, critically evaluate my work, and communicate insights effectively aligns with both the Unit Learning Outcomes and Graduate Learning Outcomes, evidencing a level of performance that meets the expectations of a High Distinction.

Checklist	Included
Learning Summary Report	✔
All tasks required for the target grade completed	✔
Evidence of any additional task(s) or activities completed	✔

Declaration

Learning Summary Report

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: 

Portfolio Overview

This portfolio demonstrates comprehensive achievement of all Unit Learning Outcomes (ULOs) for SIT772- Database Fundamentals to a High Distinction level. It showcases a balanced progression across theoretical understanding, hands-on database work, and research-based critical thinking. A key highlight of my portfolio is the Task 3.2HD research report, where I investigated the integration of machine learning (ML) and data systems, offering insights that significantly deepened my understanding of modern database practices and their intersection with intelligent systems.

Over the course of SIT772- Database Fundamentals, I have progressed from foundational knowledge of databases to mastering advanced skills in relational design, SQL implementation, procedural programming, and research-informed data architecture. My portfolio demonstrates not only the successful application of all five Unit Learning Outcomes (ULOs) but also clear evidence that I have extended well beyond the unit material, reflecting the standard of a High Distinction.

I began the trimester with basic familiarity with databases- comfortable with terms like "tables" and "queries" but lacking practical experience in design, normalization, or system integration. Through the early tasks, I built a strong theoretical foundation by engaging critically with the concepts of data, information, and knowledge, and understanding the limitations of traditional file systems in contrast to relational databases. This understanding shaped the way I approached every subsequent task: not as isolated exercises, but as components of a larger data management ecosystem.

A key turning point was learning to analyse business requirements and model them effectively using Entity Relationship Diagrams. I applied normalization techniques confidently- up to Third Normal Form- to design efficient, scalable schemas that reduced redundancy and preserved data integrity. The real-world complexity of these tasks challenged me, especially in identifying dependencies and structuring many-to-many relationships, but I overcame them through iterative refinement and targeted self-learning.

I advanced rapidly in SQL, moving from basic SELECT queries to complex multi-table JOINs, nested subqueries, data transformation, and constraint enforcement. I implemented full relational databases using MySQL and Oracle, demonstrating fluency in both environments. My work with PL/SQL- including stored procedures, triggers, and functions- went beyond the expected learning outcomes and allowed me to simulate realistic business logic within the database environment.

Where I truly believe I exceeded expectations was in my research report on the intersection of machine learning and modern database systems. This project was self-driven, involving academic literature, industry white papers, and practical case studies. I investigated cutting-edge topics such as in-database model serving, scalable data pipelines (ETL vs ELT), federated learning, and ML-aware data governance. By critically evaluating the infrastructural needs of intelligent systems, I gained a perspective that allowed me to reframe my database work- not as standalone design- but as the foundation of future-ready AI ecosystems.

In addition, I extended my learning through data visualization and BI tools by building PivotCharts in Excel and designing a data-driven form interface using web technologies and embedded SQL. These tasks, though not central to the core unit, demonstrate my initiative to explore integration points between databases and user interaction, and to think holistically about the end-to-end data lifecycle.

Throughout the trimester, I approached each task with the mindset of continuous improvement. I actively engaged with feedback, revised submissions, and sought out advanced resources to refine my understanding. I overcame challenges in form validation, relational modelling, and PL/SQL debugging not through trial and error alone, but by drawing on documentation, peer collaboration, and external tutorials- demonstrating both resilience and self-direction.

Each task in this portfolio aligns strongly with the Unit Learning Outcomes:

- ULO1: Demonstrated through foundational knowledge and critical comparison of traditional and modern data systems.
- ULO2: Applied through professional ERD design and normalization grounded in real-world scenarios.
- ULO3: Evidenced by robust, multi-table database implementation using SQL and procedural logic in PL/SQL.
- ULO4: Exceeded through original research on machine learning infrastructure and advanced data architecture.
- ULO5: Reflected in self-assessment, integration of feedback, and thoughtful synthesis across tasks.

I believe this portfolio reflects not only technical excellence and comprehensive coverage of the unit learning outcomes, but also the curiosity, discipline, and critical engagement expected at the High Distinction level. I have extended beyond the material, demonstrated initiative in bridging database practice with emerging technologies, and maintained a high standard of academic and professional integrity throughout.

I respectfully submit this portfolio as evidence of my capability, commitment, and readiness to contribute meaningfully to the data and technology landscape- and ask that it be considered for a High Distinction.

Reflections

The most important things I learnt:

This unit taught me that designing, implementing, and managing a database is not just a technical process but a structured and strategic discipline. I learnt how to approach real-world problems methodically- starting with identifying data requirements, developing normalized data models, implementing relational databases, and finally applying SQL for data manipulation and control. Importantly, I learnt to view databases not just as static systems but as dynamic components of larger architectures, particularly when integrated with business intelligence or machine learning systems. This perspective shift was a key takeaway that exceeded my original expectations.

I feel I learnt these topics, concepts, and/or tools really well:

I now feel confident in data normalization (1NF, 2NF, 3NF), writing complex SQL queries involving multiple joins and subqueries, designing ERDs, and implementing constraints and relationships in a relational schema. I also became proficient with tools like MySQL Workbench, Oracle PL/SQL, and Excel PivotTables. Moreover, I feel comfortable explaining the logic and trade-offs behind data modelling decisions, which has improved both my technical skills and communication confidence.

I found the following topics particularly challenging:

The most challenging part of the unit was working with procedural SQL in Oracle during Task 8.2D. Setting up the Oracle environment, understanding the documentation, and implementing stored procedures, triggers, and functions all presented a steep learning curve. Initially, I found the syntax complex, the error messages unclear, and the logic difficult to debug.

However, through perseverance, repeated practice, and actively seeking external resources, I gradually developed confidence and clarity. I now have a solid understanding of how PL/SQL operates and how procedural logic can be embedded within database systems. More importantly, this challenge taught me a lot about my learning process. I discovered that I learn best through hands-on experimentation, breaking down complex tasks into smaller parts, and staying resilient even when progress feels slow. Overcoming this hurdle gave me a renewed sense of confidence in my ability to tackle unfamiliar technologies independently.

I found the following topics particularly interesting

One of the most interesting and valuable parts of this unit was learning how to embed SQL logic into an interactive user interface, as I did in Task 9.2D. Designing a Windows Forms application that connects to a MySQL database helped me understand how database systems operate in real-world software environments- beyond writing queries in a console or editor.

It was fascinating to see how input validation, constraint handling, and SQL insertion commands come together in a user-facing application. Building the form from scratch, handling events, and using parameterized SQL to prevent injection gave me a deeper appreciation of the security and usability considerations involved in data systems. More importantly, this task bridged my database skills with my interest in software development, reinforcing my understanding of how front-end and back-end systems integrate.

This task helped me realize how much I enjoy building practical tools that interact with databases- something I hadn't fully appreciated before. It sparked my interest in pursuing more work in full-stack development or data-driven applications, where I can design solutions that are both technically sound and user-focused.

I still need to work on the following areas:

While I have gained a strong foundation in relational database systems, I recognize there are still areas where I can improve and grow further. In particular, I want to deepen my understanding of performance optimization techniques- such as indexing strategies, query execution plans, and load balancing in multi-user environments. These are advanced topics that I encountered conceptually but have not yet fully explored in practice.

I also identified that I could improve my efficiency in handling unfamiliar environments. Tasks involving Oracle PL/SQL and system setup required considerable time navigating documentation and troubleshooting configuration issues. In future units, I aim to become more confident and systematic in approaching new technologies by building a stronger foundation in reading technical documentation and practicing with more diverse platforms.

Finally, from a personal development perspective, I intend to continue refining my time management and task estimation skills. While I maintained consistent progress overall, I found that allocating more time for extended tasks- such as research and implementation projects- would allow for deeper reflection and iteration.

As a next step, I plan to build on the knowledge gained in this unit by exploring cloud-based data systems, big data platforms, and data-driven application development, aligning with my long-term interest in intelligent systems and data engineering. This unit has provided the confidence and motivation to continue growing as a learner and aspiring IT professional.

The things that helped me most were:

The most valuable resources throughout this unit were a combination of structured learning materials, academic support, and self-directed exploration. I regularly reviewed the lecture slides on Deakin Cloud, which provided concise and well-organized summaries of core concepts. These slides were especially helpful when revisiting complex topics like normalization, transaction control, and data security, and gave me a reliable reference throughout the trimester. The workshop walkthroughs and worked examples made abstract ideas- such as ERD design, relational constraints, and advanced SQL queries- more accessible and easier to apply.

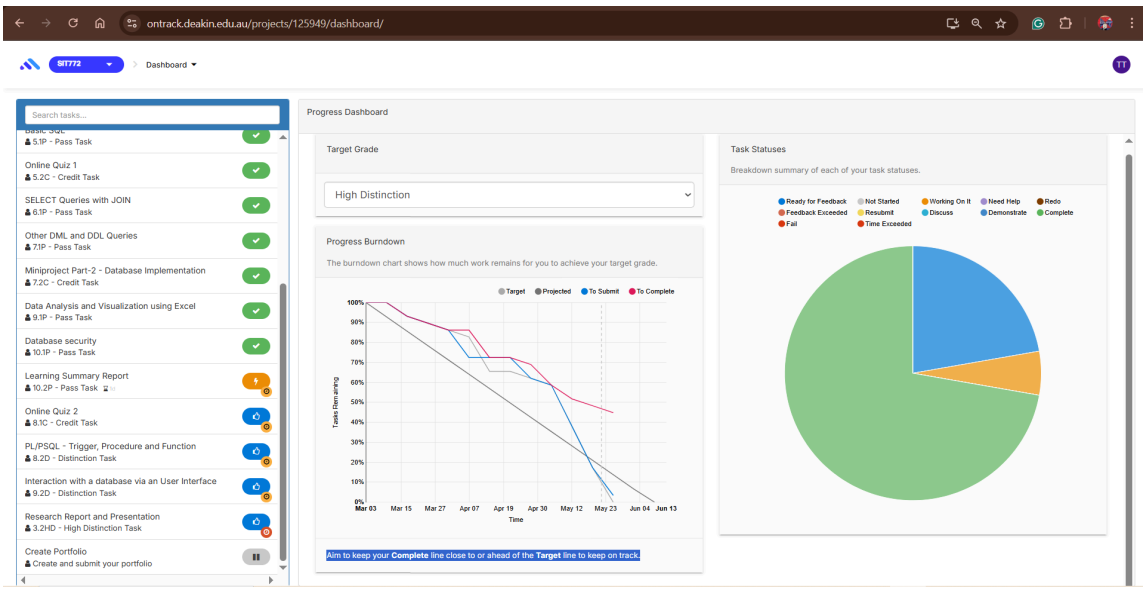
The OnTrack platform further supported my progress with clear weekly task details and supported me to improve my technical skills and improve deadline management.

Beyond this, the lecturer's explanations delivered through classes played a crucial role in helping me grasp concepts like functional dependencies and referential integrity. Concepts were often contextualized using relatable, real-world analogies, which helped bridge theory with practical application.

I also benefited from the tutoring sessions, where I had opportunities to ask questions, resolve issues, and gain timely feedback. These sessions boosted my confidence, especially when working through Oracle PL/SQL and during the independent research component of my HD task.

Finally, learning to navigate official documentation (such as MySQL and Oracle references) and engaging with academic and industry sources during my research project helped me think more critically and become a more independent learner. It was this combination of flexible cloud-based learning, strong academic guidance, and self-driven inquiry that made the unit both effective and rewarding.

My progress in this unit was …:



My progress in this unit was steady, intentional, and goal-focused. The OnTrack burndown chart reflects a consistent commitment to completing tasks over time, with all tasks completed well before the deadline. I maintained a trajectory that closely followed the projected target line, which shows that I took a proactive approach to managing my workload rather than rushing at the last minute.

By consistently aiming to stay ahead of schedule, I gave myself the space to revise, reflect, and seek feedback- especially on complex or distinction-level tasks such as the PL/SQL implementation (8.2D), embedded user interface (9.2D), and the research report and presentation (3.2HD). This approach allowed me to explore the material more deeply and extend my learning beyond the unit expectations.

The progress pie chart further shows that the majority of my tasks were completed successfully, with very few remaining in "pending" status for marking. This balance reflects not only effective time management but also a clear strategy in how I tackled progressively more advanced tasks as the unit developed. Overall, the progress data reinforces my ability to self-manage, plan ahead, and engage meaningfully with the learning process- all qualities that support my application for a High Distinction.

If I did this unit again, I would do the following things differently:

Reflecting on this unit, one thing I would do differently is to seek more intentional and meaningful engagement in class and offer support to peers when I could, I found that these efforts sometimes felt surface-level or didn't lead to deeper insight. If I had approached these moments with more focus- such as preparing thoughtful questions, offering peer support that challenged both of us to think critically, or initiating discussions about advanced topics- I believe I could have gained more from the collective learning experience.

I also extended my learning beyond the unit, particularly through my research project, which connected machine learning and database infrastructure. However, looking back, I now realise I could have gone further by applying those insights in more practical ways, such as experimenting with cloud-based ML pipelines or prototyping integration with database systems. The learning was there- but the impact could have been stronger had I pushed myself to apply it more boldly.

This unit helped me see that it's not just about what you do beyond the core requirements, but how deeply and intentionally you do it. In future units, I want to focus less on the quantity of extra effort and more on the quality and depth of engagement- whether that's through stronger contributions in class, more impactful peer collaboration, or more applied experimentation outside of assessments.

Other…:

One of the most valuable outcomes of this unit has been the shift in how I see myself- not just as a student completing tasks, but as a growing IT professional who builds systems with real-world purpose, long-term impact, and ethical responsibility. Through designing normalized data models, writing secure and structured SQL solutions, and engaging in

hands-on implementation, I've developed a deeper understanding of how databases operate at the heart of modern information systems.

This growth was especially solidified through my High Distinction research project, where I explored the integration of machine learning and database infrastructure. That task pushed me far beyond the standard expectations, requiring me to critically evaluate tools like MLflow and BigQuery, assess architectural trade-offs, and reflect on emerging concerns like data drift, privacy, and ethical model deployment. It strengthened my ability to work independently, synthesise academic and industry sources, and contribute to conversations about the future of intelligent systems.

This unit also reinforced the importance of aligning technical skills with broader values: data ethics, scalability, usability, and maintainability. I learned that successful systems aren't just functional- they are sustainable, adaptable, and thoughtfully designed to meet evolving needs. Whether I was writing a PL/SQL trigger, building a WinForms-based embedded SQL interface, or supporting peers with explanations, I was constantly encouraged to think critically about both how and why we build systems the way we do.

Perhaps most importantly, SIT772 reminded me that meaningful growth comes not only from meeting requirements, but from the choices we make to explore beyond them. The unit gave me confidence in my ability to independently learn new tools and frameworks, and a mindset focused on continuous learning, real-world relevance, and thoughtful contribution. These lessons have helped define my professional direction, and I carry them forward with clarity and purpose into future roles in data engineering, full-stack development, or intelligent systems architecture.

21 Interaction with a database via an User Interface

Desgin an user interface form to populate and display required results from a database and prepare a video demonstration

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical database concepts and real-world application development, enhancing both my technical and problem-solving skills.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical database concepts and real-world application development, enhancing both my technical and problem-solving skills.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical database concepts and real-world application development, enhancing both my technical and problem-solving skills.

Outcome	Weight
Reflection	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical database concepts and real-world application development, enhancing both my technical and problem-solving skills.

Outcome	Weight
Research and critical review	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical

database concepts and real-world application development, enhancing both my technical and problem-solving skills.

Date	Author	Comment
2025/05/15 15:48	Thi Ngo	Working On It
2025/05/20 16:55	Thi Ngo	Ready to Mark
2025/05/20 17:10	Thi Ngo	Dear Tutor,I am writing to submit my video presentation for Task 9.2D – Embedded SQL Form Implementation. The video includes an explanation of the technologies used, a demonstration of the form interface, and examples of both unsuccessful and successful data insertions into the database.You can view the video at the following link::link: https://www.canva.com/design/DAGnzMmzUdU/VcmEq_6Eh2SbOVIAh9upFA/view?utm_content=DAGnzMmzUdU&utm_campaign=design-share&utm_medium=link&utm_source=recording_viewPlease let me know if there are any issues accessing the video.Kind Regards,Suong Ngo.
2025/05/22 10:46	Mohammad Hossain	Looks good. However, in your video demonstration unsuccessful insertion attempt is missing while you have provided the screenshot in the report. Please update your recording accordingly.
2025/05/22 10:46	Mohammad Hossain	your report is missing in-text citation as well as a list of references.
2025/05/22 10:46	Mohammad Hossain	Fix and Resubmit
2025/05/22 12:10	Thi Ngo	Working On It
2025/05/22 12:10	Thi Ngo	Ready to Mark
2025/05/22 12:16	Thi Ngo	Dear Dr. Mohammad,Thank you for your feedback. I have updated the report to include the required in-text citations and a list of references. Additionally, due to an issue with the Canva video demonstration, I have re-recorded the video and uploaded it to YouTube instead. You can view it here::link: https://youtu.be/uNot5td5HycThis updated video now includes the previously missing demonstration of an unsuccessful insertion attempt, which was already shown in the report screenshots.Please let me know if anything else needs adjustment.Kind Regards,Suong Ngo.
2025/05/22 13:50	Mohammad Hossain	It looks good now. Thanks.
2025/05/22 13:50	Mohammad Hossain	Complete

Deakin University

Database Fundamentals

OnTrack Submission

Interaction with a database via an User Interface

Submitted By:
Thi Thu Suong Ngo
s224757434
2025/05/22 12:10

Tutor:
Mohammad belayet Hossain

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Through this task, I gained practical experience in building a user-friendly form interface and integrating it with a relational database using embedded SQL. I learned how to validate user input effectively, handle errors gracefully, and enforce data integrity by aligning front-end validation with backend constraints. Additionally, I deepened my understanding of parameterized queries for secure SQL execution and exception handling in C#. This task helped me bridge the gap between theoretical database concepts and real-world application development, enhancing both my technical and problem-solving skills.

May 22, 2025

1. Overview

This report documents the design and implementation of a Windows Forms application developed in C#. Its core function is to insert records into the USER table of a MySQL database. The primary objective was to demonstrate SQL integration within a user-friendly graphical interface, while applying principles of input validation, exception handling, and secure database interactions through parameterized queries (Troelsen & Japikse, 2021).

2. Technologies and Tools Used

Component	Technology/Tool
Programming Language	C# (.NET)
UI Framework	Windows Forms (WinForms)
Database System	MySQL 8.0
Connector	MySql.Data NuGet package
IDE	Visual Studio Code
Application Type	Desktop Application (WinForms)

Windows Forms was chosen for its ease of use in creating event-driven interfaces in C#, while MySQL was selected for its reliability and widespread adoption as a relational database system (Elmasri & Navathe, 2017).

3. Selected Table: USER

The application was designed to interface with the USER table, structured as follows:

Column	Data Type	Constraint
USER_ID	INT	PRIMARY KEY
NAME	VARCHAR(100)	NOT NULL
EMAIL	VARCHAR(100)	NOT NULL, UNIQUE
PASSWORD	VARCHAR(100)	NOT NULL
COUNTRY	VARCHAR(50)	Optional
JOIN_DATE	DATE	Format: YYYY-MM-DD

IS_ACTIVE	BOOLEAN	Default TRUE

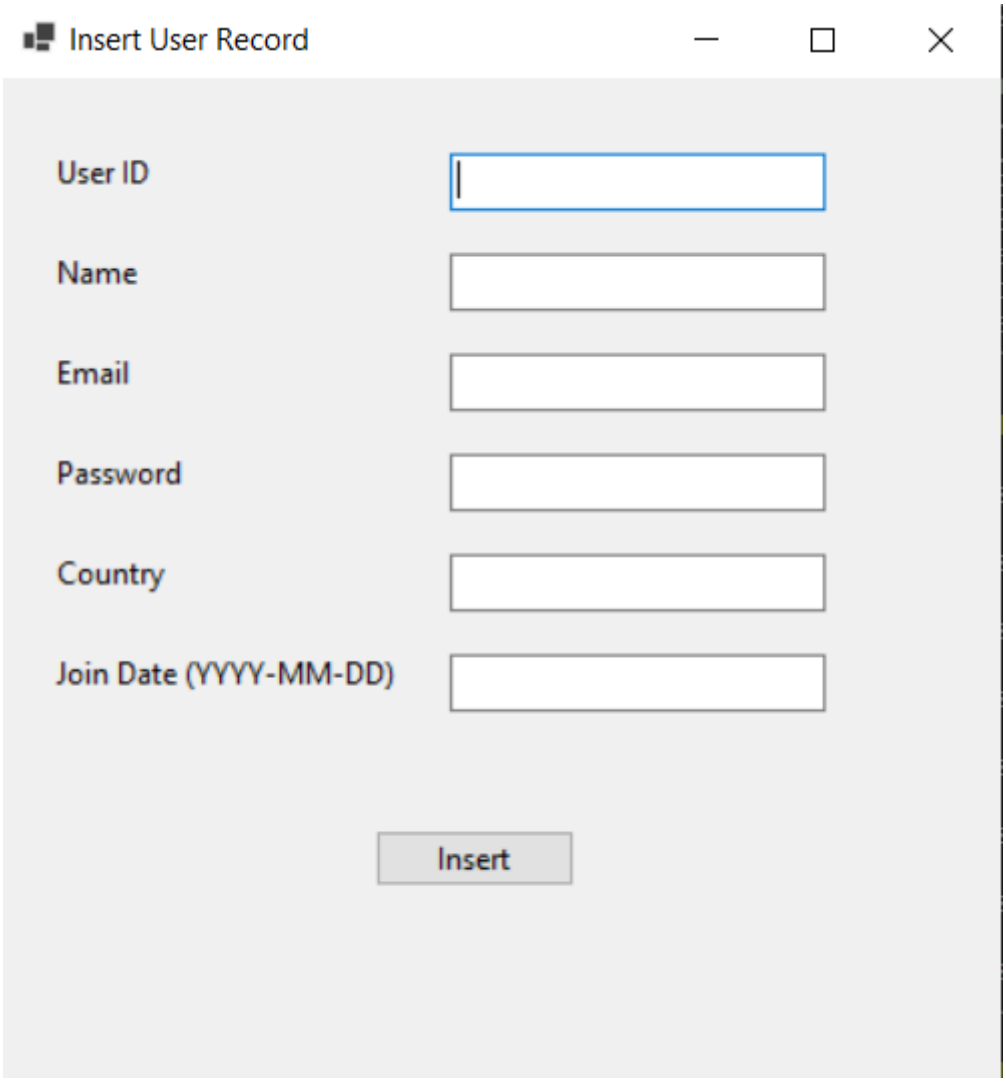
These constraints follow best practices in database normalization and integrity enforcement (Connolly & Begg, 2015).

4. Features Implemented

- A user interface with 6 textboxes and an "Insert" button
- Input validation:
    - Checks for empty required fields
    - Validates data types (e.g., USER_ID as INT, JOIN_DATE as DATE)
    - Validates email format
- Constraint handling:
    - Checks for duplicate USER_ID or EMAIL before insert
- Confirmation and error messages
- Database insertion using parameterized SQL to prevent SQL injection

5. Screenshots

- **UI of the form**



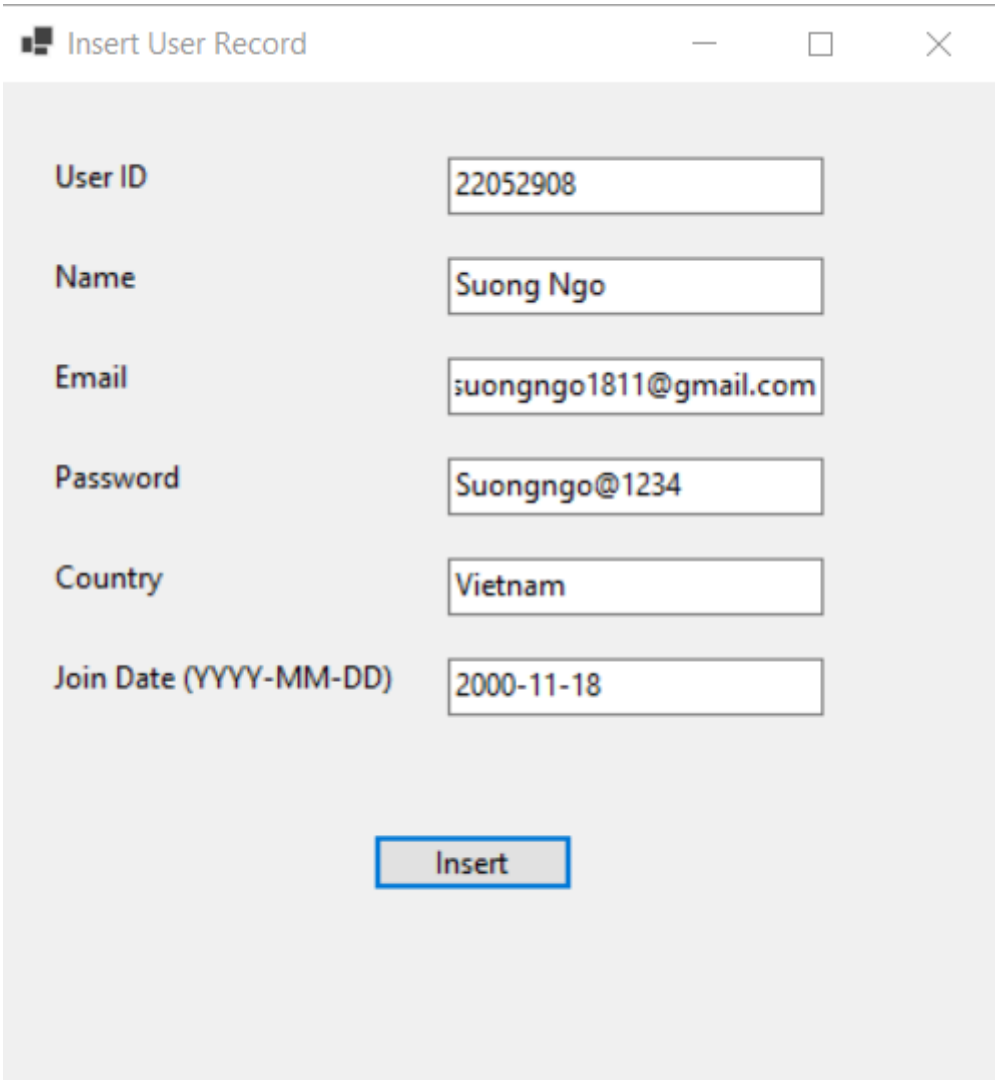
- Insertion and MySQL Workbench showing new row in user table

- **Check table before creating User Table Form:**

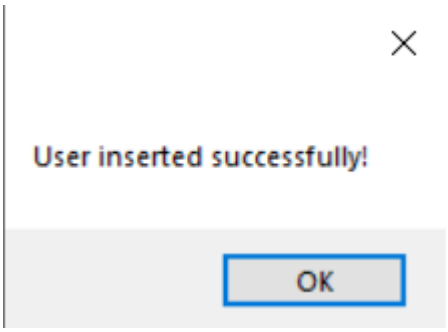
SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE	IS_ACTIVE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01	1
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20	1
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15	1
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10	1
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Insert user**



- **Insert User Confirmation**

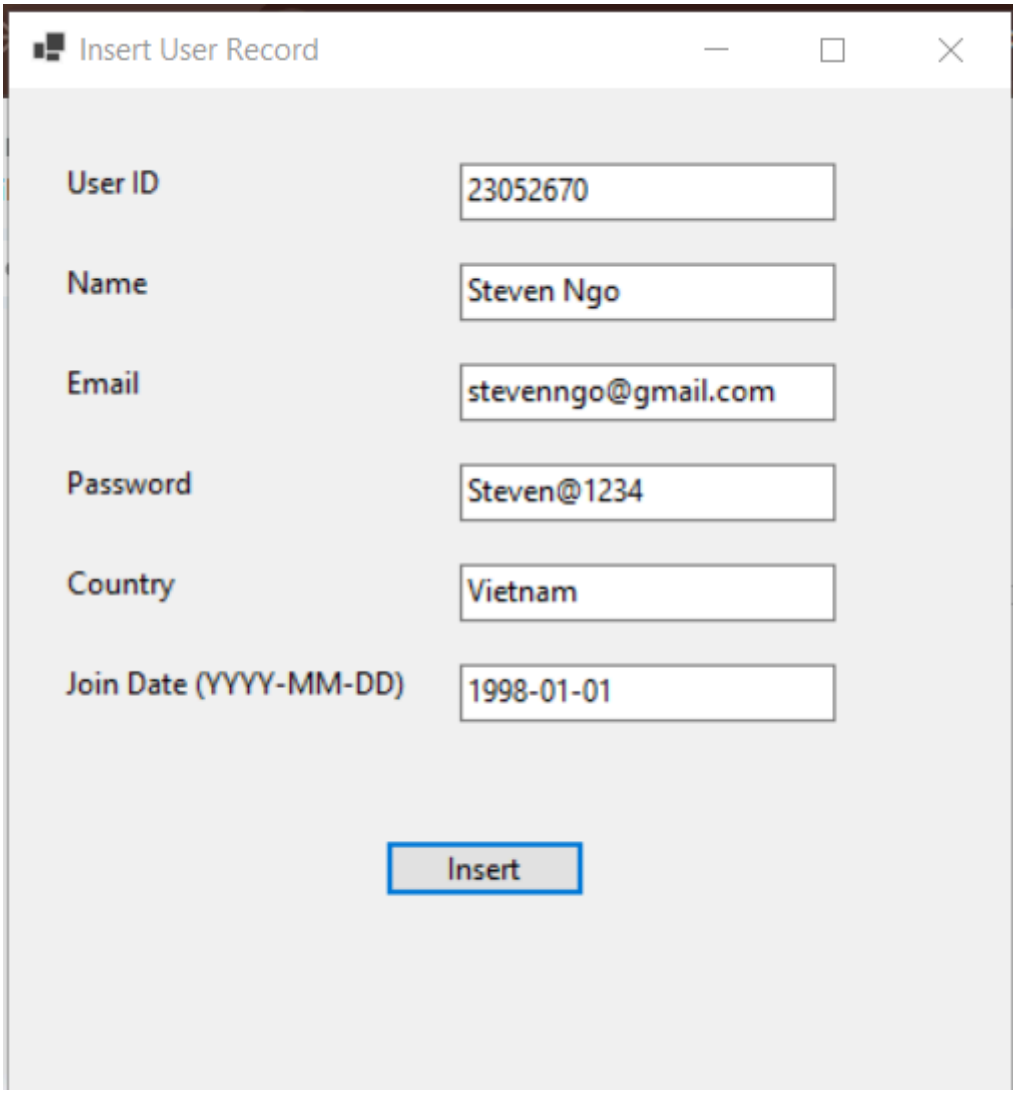


- **Check table after creating User Table Form:**

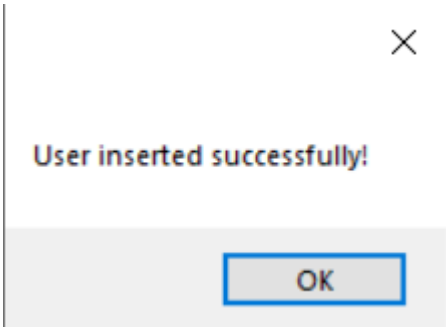
SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE	IS_ACTIVE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01	1
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20	1
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15	1
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10	1
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30	0
22052908	Suong Ngo	suongngo1811@gmail.com	Suongngo@1234	Vietnam	2000-11-18	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Insert another value:**



- **Insertion Confirmation**



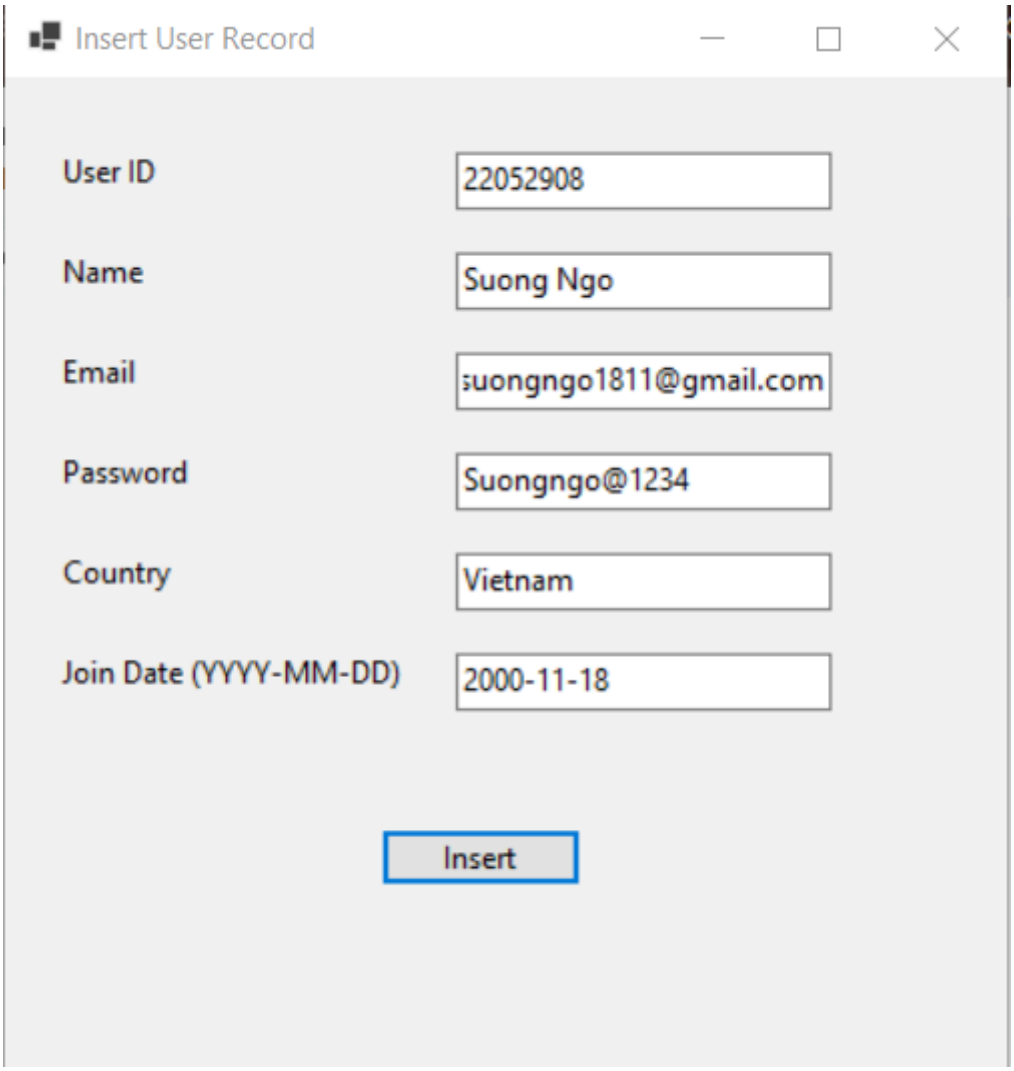
- **Check table after creating User Table Form:**

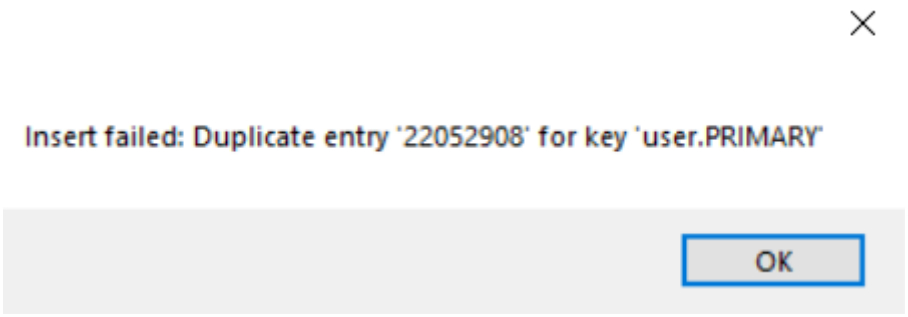
SELECT * FROM USER;

USER_ID	NAME	EMAIL	PASSWORD	COUNTRY	JOIN_DATE	IS_ACTIVE
201	Nguyen Tuan	tuan@gmail.com	pass123	Vietnam	2024-01-01	1
202	James Walker	james@gmail.com	walker456	Australia	2023-12-20	1
203	Hiro Tanaka	hiro@gmail.com	tanaka789	Japan	2023-11-15	1
204	Lily Chen	lily@gmail.com	lilyabc	China	2024-02-10	1
205	Sarah Kim	sarahk@gmail.com	sarah321	South Korea	2023-10-30	0
22052908	Suong Ngo	suongngo1811@gmail.com	Suongngo@1234	Vietnam	2000-11-18	1
23052670	Steven Ngo	stevenngo@gmail.com	Steven@1234	Vietnam	1998-01-01	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

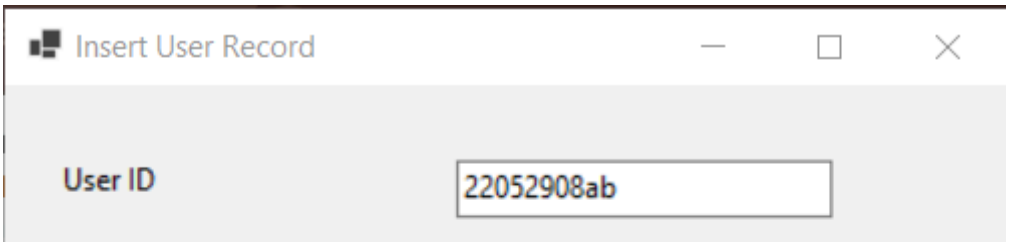
- Error message for invalid input

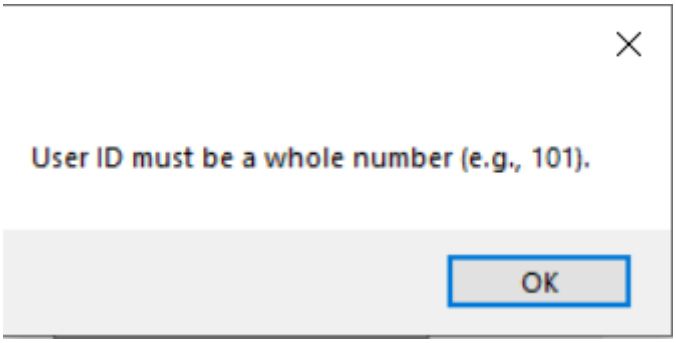
As I try to insert duplicate value, it shows duplicate error:



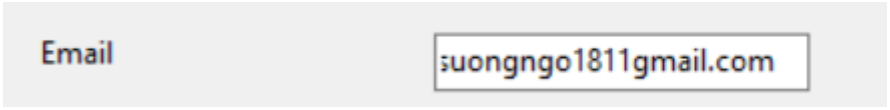


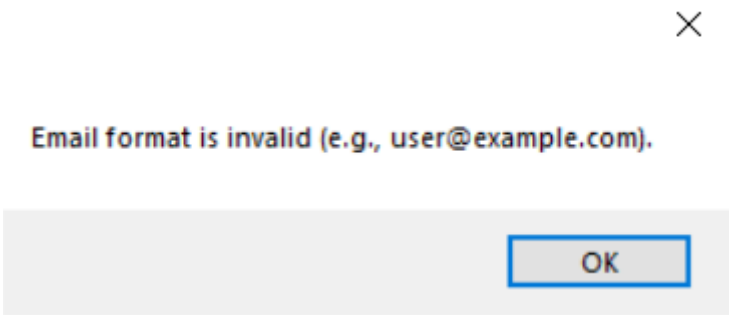
- **UserID Type Validation:**



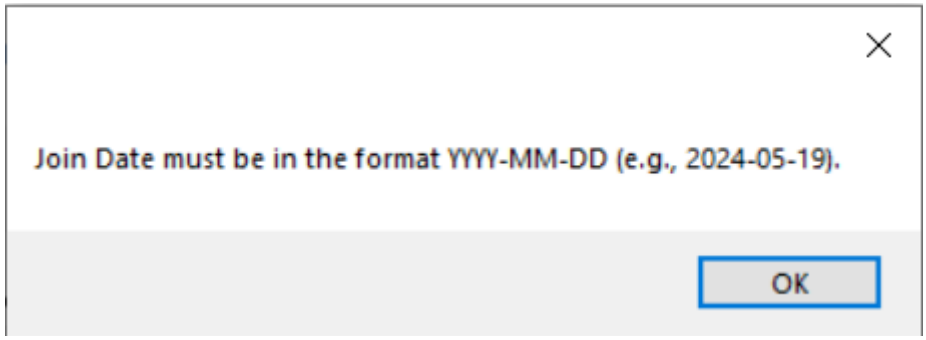


- **Email Type Validation**





- **Joint Date Validation**



6. Code Explanation

The implementation of this Windows Forms application consists of three key source files: Program.cs, Form1.cs, and Form1.Designer.cs. Each plays a distinct role in the structure and behavior of the application, contributing to the user interface, control flow, and integration with the MySQL database

- Program.cs: Entry Point of the Application

- **Purpose:** Contains the Main method that initializes the application interface using Application.Run(new Form1()). This is a standard method of launching GUI applications in .NET (Troelsen & Japikse, 2021).
- **Key Code:**

```
Application.Run(new Form1());
```

- **Explanation:** The Program.cs file is responsible for setting up the application window. It applies standard UI settings using Application.EnableVisualStyles() and initializes the rendering behavior via Application.SetCompatibleTextRenderingDefault(false). The line Application.Run(new Form1()) loads the main form (Form1) that contains the user interface. This file typically does not require modifications unless a different form is used as the starting interface.

- Form1.cs: Core Logic + Event Handling

- **Purpose:** This is the main file that controls the application's logic, including user interface creation, event handling, input validation, and interaction with the MySQL database. This structure supports modularity and aligns with object-oriented principles such as abstraction and encapsulation (C# Programming Guide, Microsoft Docs, 2023).
- **Key Features:**
    - Constructs the interface using Label, TextBox, and Button components programmatically (without drag-and-drop).

    - Implements event handling for the “Insert” button, allowing user data to be submitted.

    - Validates critical input fields such as USER_ID (ensuring numeric input), JOIN_DATE (requiring a valid date format), and EMAIL (format and uniqueness).

    - Connects to the MySQL database (w8sit772) using a secure connection string and the MySql.Data library.

    - Executes parameterized SQL INSERT queries via MySqlCommand to avoid SQL injection vulnerabilities.

    - Handles exceptions gracefully and provides appropriate feedback through MessageBox alerts.
- **Explanation:** Form1.cs encapsulates the functional behavior of the form, combining both object-oriented programming principles and database interaction. It demonstrates method abstraction (e.g., separating validation logic) and encapsulation by keeping control-related logic within the form class. This file is central to fulfilling the core requirement of embedding SQL into a user-driven interface.

- **Form1.Designer.cs: UI Initialization Structure**

- **Purpose:** This file is auto-generated when using Visual Studio’s design tools but remains essential even when form components are created manually. It provides the definition for the InitializeComponent() method, which is invoked during form construction.

```
InitializeComponent();
```

- **Key Role:**

    - Defines and initializes base form properties such as size, name, and events.

    - Contains the InitializeComponent() method that is automatically executed when Form1 is instantiated.

    - Ensures compatibility with the Windows Forms framework even when manual control creation is preferred.

- **Explanation:** Although this implementation builds the form programmatically in Form1.cs, the presence of Form1.Designer.cs is still required to maintain proper structure within the WinForms framework. The file can remain minimal, but it is a necessary component for application lifecycle management.

7. All Code

- **Program.cs**

```
// Source: Microsoft Docs, 2023 - Entry point for C# WinForms app
using System;
using System.Windows.Forms;

namespace UserFormApp
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles(); // Enables visual styles
(Microsoft Docs, 2023)
            Application.SetCompatibleTextRenderingDefault(false); //
Standard rendering setup
            Application.Run(new Form1()); // Launches the Form1 UI
        }
    }
}
```

- **Form1.Designer.cs**

```
// Designer code generated by Visual Studio (Microsoft Docs, 2023)

namespace UserFormApp
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up resources.
        /// </summary>
        protected override void Dispose(bool disposing)
```

```
        {
            if (disposing && (components != null)) components.Dispose();
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        private void InitializeComponent()
        {
            this.SuspendLayout();
            //
            // Form1
            //
            this.ClientSize = new System.Drawing.Size(400, 400);
            this.Name = "Form1";
            this.Load += new System.EventHandler(this.Form1_Load);
            this.ResumeLayout(false);
        }

        #endregion
    }
}
```

- **Form1.cs**

```
// This code demonstrates form design, event handling, and MySQL
connection with parameterized queries


using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient; // MySQL .NET Connector (Oracle, 2024)

namespace UserFormApp
{
    public partial class Form1 : Form
    {
        private TextBox txtUserId, txtName, txtEmail, txtPassword,
txtCountry, txtJoinDate;
```

```
        private Button btnInsert;

        // UPDATE this with MySQL root password
        // Standard MySQL connection string (Oracle, 2024)
      private string connStr =
"server=localhost;port=3306;user=root;password=Sophiengo@1234;database=w8s
it772;";

        public Form1()
        {
            InitializeComponent();
            CreateFormUI(); // Builds the form programmatically (Troelsen
& Japikse, 2021)
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void CreateFormUI()
        {
            this.Text = "Insert User Record";

            Label[] labels = {
                new Label() { Text = "User ID" },
                new Label() { Text = "Name" },
                new Label() { Text = "Email" },
                new Label() { Text = "Password" },
                new Label() { Text = "Country" },
                new Label() { Text = "Join Date (YYYY-MM-DD)" }
            };

            TextBox[] textBoxes = {
                txtUserId = new TextBox(),
                txtName = new TextBox(),
                txtEmail = new TextBox(),
                txtPassword = new TextBox(),
                txtCountry = new TextBox(),
                txtJoinDate = new TextBox()
```

```
            };

            for (int i = 0; i < labels.Length; i++)
            {
                labels[i].Top = 30 + i * 40;
                labels[i].Left = 20;
                labels[i].Width = 150;
                this.Controls.Add(labels[i]);

                textBoxes[i].Top = labels[i].Top;
                textBoxes[i].Left = 180;
                textBoxes[i].Width = 150;
                this.Controls.Add(textBoxes[i]);
            }

            btnInsert = new Button() { Text = "Insert", Top = 300, Left =
150, Width = 80 };
            btnInsert.Click += BtnInsert_Click; // Event handler binding
(Microsoft Docs, 2023)
            this.Controls.Add(btnInsert);
        }

        private void BtnInsert_Click(object sender, EventArgs e)
        {
            if (!ValidateInput())
            {
                MessageBox.Show("Please fill all required fields!");
                return;
            }

            try
            {
                using (MySqlConnection conn = new
MySqlConnection(connStr))
                {
                    conn.Open();
                    string query = "INSERT INTO USER (USER_ID, NAME,
EMAIL, PASSWORD, COUNTRY, JOIN_DATE) " +
                                   "VALUES (@user_id, @name, @email,
@password, @country, @join_date)";
```

```
                    using (MySqlCommand cmd = new MySqlCommand(query,
conn))
                    {
                    // Parse and validate types
                    // Defensive programming: validate types and formats
before insertion (Halfond et al., 2006)


                    int userId;
                    DateTime joinDate;

                    if (!int.TryParse(txtUserId.Text, out userId))
                    {
                        MessageBox.Show("User ID must be a whole number
(e.g., 101).");
                        return;
                    }

                    if (!DateTime.TryParse(txtJoinDate.Text, out
joinDate))
                    {
                        MessageBox.Show("Join Date must be in the format
YYYY-MM-DD (e.g., 2024-05-19).");
                        return;
                    }

                    // Validate email format
                    if (!txtEmail.Text.Contains("@") ||
!txtEmail.Text.Contains("."))
                    {
                        MessageBox.Show("Email format is invalid (e.g.,
user@example.com).");
                        return;
                    }

 // Use of parameterized queries to prevent SQL injection (Halfond et al.,
2006)
                        cmd.Parameters.AddWithValue("@user_id", userId);
```

```
                        cmd.Parameters.AddWithValue("@name",
txtName.Text);
                        cmd.Parameters.AddWithValue("@email",
txtEmail.Text);
                        cmd.Parameters.AddWithValue("@password",
txtPassword.Text);
                        cmd.Parameters.AddWithValue("@country",
txtCountry.Text);
                        cmd.Parameters.AddWithValue("@join_date",
joinDate);

                        cmd.ExecuteNonQuery();
                        MessageBox.Show("User inserted successfully!");
                    }
                }
            }
            catch (MySqlException ex)
            {
                MessageBox.Show("Insert failed: " + ex.Message);
            }
        }

        private bool ValidateInput()
        {
            return !string.IsNullOrWhiteSpace(txtUserId.Text) &&
                   !string.IsNullOrWhiteSpace(txtName.Text) &&
                   !string.IsNullOrWhiteSpace(txtEmail.Text) &&
                   !string.IsNullOrWhiteSpace(txtPassword.Text) &&
                   !string.IsNullOrWhiteSpace(txtJoinDate.Text);
        }
    }
}
```

References

Connolly, T. & Begg, C., 2015. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Harlow: Pearson Education.

Elmasri, R. & Navathe, S.B., 2017. *Fundamentals of Database Systems*. 7th ed. Boston: Pearson.

Halfond, W.G.J., Viegas, J. & Orso, A., 2006. A classification of SQL-injection attacks and countermeasures. *Proceedings of the IEEE International Symposium on Secure Software Engineering*, pp.13–15.

Troelsen, A. & Japikse, P., 2021. *Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming*. 10th ed. Berkeley: Apress.

Halfond, W.G.J., Viegas, J. and Orso, A., 2006. A classification of SQL-injection attacks and countermeasures. *Proceedings of the IEEE International Symposium on Secure Software Engineering*, pp.13–15.

Microsoft Docs, 2023. *Windows Forms Overview*. [online] Available at: https://learn.microsoft.com/en-us/dotnet/desktop/winforms/ [Accessed 22 May 2025].

Microsoft Docs, 2023. *C# Programming Guide*. [online] Available at: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/ [Accessed 22 May 2025].

Oracle, 2024. *MySQL Connector/NET Developer Guide*. [online] Available at: https://dev.mysql.com/doc/connector-net/en/ [Accessed 22 May 2025].

Troelsen, A. and Japikse, P., 2021. *Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming*. 10th ed. Berkeley: Apress.